Ecole Normale Supérieure de Lyon

Applications of Structure-Preserving Cryptography and Pairing-Based NIZK Proofs

Benoît LIBERT

Chercheur

Mémoire d'habilitation à diriger des recherches

Présenté le 29 mai 2015 après avis des rapporteurs :

Charanjit JUTLA, Researcher, IBM Research (USA)
Eike KILTZ, Professor, Ruhr University of Bochum (Germany)
David POINTCHEVAL, Directeur de recherche, CNRS, École Normale Supérieure

Examinateurs :

Dario CATALANO, Professor, Università di Catania (Italy)
Guillaume HANROT, Professeur, Ecole Normale Supérieure de Lyon
Pascal PAILLIER, CEO and Senior Security Expert at CryptoExperts, Paris
David POINTCHEVAL, Directeur de recherche, CNRS, École Normale Supérieure
Brigitte VALLÉE, Directrice de recherche, CNRS, Université de Caen

Contents

Introduction

This document presents some of the results I obtained in the recent years in the area of cryptography.

My current and past researches were devoted to the design of efficient and provably secure public-key cryptographic schemes. These days, when it comes to proposing a new cryptosystem, it is (fortunately) a common practice to provide strong evidence of its security by means of a rigorous security proof. To this end, one should first formally define what it means for the specific cryptographic primitive to be secure. Then, a common approach consists in giving a reduction showing that, in the sense of the considered security definition, any efficient adversary (i.e., with polynomial running time in the security parameter) breaking the system with non-negligible probability would imply a polynomial algorithm solving a hard problem (such as factoring large integers, computing discrete logarithms, etc). The conjectured intractability of the problem in polynomial time thus implies the non-existence of polynomial adversaries. In some cases, security proofs may take place in the random oracle model [32], which is an idealized model of computation where hash functions are modeled as oracles controled by the reduction. This notably implies that, whenver the adversary wants to know the hash value of any input string, it has to ask an oracle for it and thus reveal to the reduction which hash values it decides to compute. The random oracle methodology has been subject to criticism as there are examples (see, e.g., [72]) of cryptographic schemes that have no secure instantiation with a real hash function although they do have a security proof in the random oracle model. For this reason, a security proof in the standard model (i.e., without random oracles) may be preferrable, especially when it comes at a reasonable cost. The results presented in this habilitation thesis do not rely on random oracles and thus stand in the standard model of computation.

My contributions fit within several sub-areas of public-key cryptography. In order to describe the global context of my research, these sub-areas will briefly outlined in the following pages. In this habiliation thesis, however, I will focus on the topics of anonymity-related cryptographic protocols and homomorphic cryptography, which are discussed in sections 0.5 and 0.7 of this introduction.

0.1 Identity-Based Encryption

My PhD thesis presented new applications of bilinear maps over groups where the discrete logarithm problem is presumably hard. These tools found many applications such as identity-based encryption (IBE) [236, 45], where any human-readable identifier (e.g., an email address) can serve as a public key so as to eliminate the need for digital certificates and simplify key management. The most important contribution [27] of my PhD thesis was to describe the most efficient identity-based cryptosystem combining the functionalities of signature and encryption. This research was carried out in collaboration with Paulo Barreto and Noel McCullagh.

Part of my post-doctoral research was also related to identity-based encryption. In collaboration with Damien Vergnaud, we described an improved technique [186] allowing to decrease the required amount of trust in authorities (that have to generate users' private keys and are obviously able to decrypt all ciphertexts) in IBE schemes as initially suggested by Goyal [130]. We showed [186] an efficient way to prevent dishonest authorities from re-distributing copies of users' private keys without being detected. Our technique allows tracing obfuscated decryption devices (based on their input-output behavior) that illegally decrypt users' communications back to their source. The advantage of our construction is to provide a much better efficiency than previous constructions [130, 131] enabling black-box traceability. In collaboration with Nuttapong Attrapadung, we also described [22] the first identity-based broadcast encryption scheme — where the sender can encrypt messages for several identities -– that simultaneously provides adaptive security and constant-size ciphertexts, regardless of the number of receivers. This result was published at the Public-Key Cryptography 2010 conference. Together with Nuttapong Attrapadung and Elie de Panafieu (who was an internship student of mine during the summer 2009), we also described several constructions [24, 21] of attribute-based encryption (ABE) schemes featuring short ciphertexts. In short, ABE schemes are a generalization of identity-based encryption where ciphertexts are labeled with sets of descriptive attributes whereas users' private keys encode a complex access formula specifying which ciphertexts users are entitled to decrypt. The ABE primitive is motivated by fine-grained access control over encrypted data. For example, they make it possible to selectively share one's data in cloud storage systems. Our contribution [24, 21] was to describe the first truly expressive solutions where the size of the ciphertext does not depend on the number of associated attributes.

0.2 Key-Evolving Cryptography

Between 2006 and 2009, in collaboration with Moti Yung, I explored techniques allowing to confine the effect of private key exposures – caused by hackers rather than actual cryptanalysis – within a certain time interval. With the growing use of mobile devices, it has become much easier to break into users' computer than defeating cryptosystems by solving hard problems. One way to address this concern is to update private keys at discrete time periods (without changing the public key) in such a way that the security of past periods is preserved after a key exposure. This property is termed "forward security". Our main result [181] was a generic technique allowing a computer to automatically handle key updates (without any human intervention) in forward-secure signatures where private keys are shielded by a second factor, such as a password. Most previous key-evolving signatures were not compatible with this kind of additional password-based key protection since, in straightforward implementations, users had to enter their password at each update operation, which was impractical in case of frequent updates. Our results [181, 182] consisted of generic ways allowing an untrusted computing environment to update an encrypted version of the user's private key, in such a way that passwords only come into play to sign messages.

0.3 Cryptographic Schemes with Delegation

In 2008, in collaboration with Damien Vergnaud, we studied [184] key delegation techniques that find applications in the secure forwarding of encrypted emails or in distributed file systems. As initially suggested by Blaze, Bleumer and Strauss [35], a proxy re-encryption system (PRE) is an encryption scheme where a delegator A can provide a proxy with a re-encryption key allowing to translate ciphertexts initially encrypted for A into ciphertexts encrypted for a delegatee B. The proxy should be able to do so without seeing underlying plaintexts or any user's private key. Our contribution [184] was to describe the first unidirectional PRE system (where the proxy can translate from A to B without being also able to translate from B to A) that can be proven secure against chosen-ciphertext attacks, where the adversary has access to a decryption oracle. Later on, we addressed similar problems in the context of signature schemes [183], where a proxy should be able to translate B's signatures into signatures bearing A's name. In 2005, Ateniese and Hohenberger showed how cryptographic bilinear maps can be used to design unidirectional proxy re-signatures (PRS), which are useful for the inter-domain conversion of digital certificates. They left open the problem of constructing unidirectional PRS schemes where signatures can be translated in sequence (from A to B first, then from B to C and so on). We provided the first step [183] towards efficiently solving this problem suggested for the first time by Blaze, Bleumer and Strauss in 1998 [35].

0.4 Distributed Cryptography

Threshold cryptography [96, 98] aims at avoiding single points of failure by splitting private keys into n shares, each one of which is given to a different server, in such a way that at least t of these shares should be combined to recover the original private key. This implies that at least t servers should contribute to private key operations (namely, the decryption procedure in a public-key encryption scheme and the signing process in digital signatures). A threshold primitive is said robust if a malicious adversary who corrupts at most $t-1$ servers cannot prevent the honest majority (which exists when $n \geq 2t-1$) from successfully completing their operations. Threshold cryptographic schemes have been mostly analyzed in the scenario of static corruptions, where the adversary has to choose which servers he wants to corrupt before the generation of the public key. Unfortunately, adaptive adversaries (who can choose whom to corrupt at any time, based on their complete view) turn out to be harder to deal with. In the context of robust threshold public-key encryption systems with chosen-ciphertext security (i.e., that resist adversaries equipped with a decryption oracle), most adaptively secure solutions have a relatively complex decryption protocol, where some interaction is required among decryption servers. In collaboration with Moti Yung, we proposed the first fully non-interactive robust threshold cryptosystems that provide chosen-ciphertext security against adaptive adversaries in the standard model. In 2011, we first described a scheme [189] based on specific number theoretic assumptions. In 2012, we provided a more general framework [191] for constructing such threshold cryptosystems and gave several instantiations with a better efficiency than our initial realization.

0.5 Anonymity-Related Cryptographic Primitives

Between 2009 and 2014, I also worked on privacy-enhancing cryptographic mechanisms such as those allowing users to accountably hide in a crowd. Group signatures[1], as introduced by Chaum and Van Heyst [85], allow registered members of a group to anonymously sign messages in the name of the entire group. If necessary, an authority is able to identify the signer using some secret information. This primitive finds applications in trusted computing platforms or in electronic auction systems. It is well-known how to construct efficient group signatures in the random oracle model [15] and in the standard model [55, 56, 134]. Traceable signatures [155] extend group signatures in that the group manager can additionally reveal a user-specific trapdoor allowing to publicly trace all signatures issued by a given member suspected of illegal activity. Hence, misbehaving users' signatures can be traced without requiring the opening authority to open all signatures, which would harm the privacy of honest users. In a joint work with Moti Yung [187, 190], we constructed the first efficient traceable signature scheme that does not appeal to the random oracle model.

In the area of group signatures, I also paid attention to the revocation problem, which consists in efficiently disabling the anonymous signing capability of expelled group members and only these members. Together with Damien Vergnaud [185], we proposed a first solution in the standard model in 2009. Unfortunately, this approach has the disadvantage of incurring a verification cost linear in the number of revocations. In collaboration with Moti Yung and Thomas Peters [180, 179], we subsequently showed how to avoid this limitation. Specifically, we described a new revocation mechanism which is borrowed from the literature on broadcast encryption. This approach is well-suited to group signatures in the standard model. Its main advantage over many existing solutions is that unrevoked group members do not need to update their private keys when other members are revoked. At the same time, the verification cost and the size of signatures are constant (where "constant" means that it only depends on the security parameter and not on the number of revocations or the maximal number group members). Our initial scheme improves upon a comparable mechanism (published by Nakanishi *et al.* [202]) in that it completely avoids linear complexities in the maximal cardinality of the group: the complexity is at most poly-logarithmic in all metrics. Subsequently, we further showed how to additionally obtain constant-size private keys without degrading the efficiency in other metrics.

Group encryption [156] is the encryption analogue of group signatures. Namely, a sender should be able to encrypt a message for some anonymous member of a group while appending to the ciphertext a proof that the latter is well-formed and intended for some certified group member. The primitive finds applications in the asynchronous transfer of credentials between peer devices or the verifiable encryption of keys to anonymous trusted parties. The first scheme, proposed by Kiayias, Tsiounis and Yung in 2007 [156], requires interactive conversations (at least if one is willing to avoid the random oracle model) between the sender and the proof verifier. The need for interaction is a limitation since it requires senders to be online at the same time as verifiers and to remember the random numbers that were used to encrypt all ciphertexts. In collaboration with Julien Cathalo and Moti Yung [81], we showed the first truly non-interactive scheme (i.e., no interaction is ever needed between the sender and the verifier) with a security proof in the standard model. In the same article on non-interactive group encryption [81], we described one of the first realizations (actually, the first

[1] Note that, here, the term "group" refers to a population for users rather than an algebraic structure.

practical one with a security proof under non-interactive number theoretic assumptions) of a primitive initially suggested by Groth [133] and that was subsequently called "structure-preserving signature" in the literature [4, 6]. Structure-preserving signatures are signature schemes where messages and public keys only consist of elements of an abelian group over which a bilinear map is efficiently computable. They have many applications in privacy-preserving protocols because they are fully compatible with the Groth-Sahai non-interactive proof systems [138]. The reason is that Groth-Sahai proofs can only serve as proofs of knowledge – in the sense that a knowledge extractor can recover the witnesses from any valid proof – when the witnesses are elements of an abelian group over which a bilinear map is efficiently computable. The useful property of structure-preserving signatures is that they precisely allow signing elements of bilinear groups without destroying their algebraic structure (in particular, without first hashing them). For example, this allows one to efficiently prove knowledge of a hidden message-signature pair, as typically done in group signature schemes. More efficient structure-preserving signatures appeared in the literature later on [4, 6, 2, 3].

In the context of group signatures, I also considered alternatives to factoring and discrete-logarithm-based solutions. In collaboration with Fabien Laguillaumie, Adeline Langlois and Damien Stehlé [166], we proposed the first group signature based on lattice hardness assumptions with logarithmic signature size in the cardinality of the group. In earlier lattice-based constructions [129, 69], the signature length was linear in the maximal number of group members.

0.6 Commitment Schemes with Special Properties

A commitment scheme is the digital analogue of a safe or a sealed envelope. Namely, whatever is in the envelope remains secret until the opening of that envelope. At the same time, the sender is bound to a unique message and cannot change his mind about the content when the envelope is sealed. Commitment schemes are a fundamental cryptographic primitive (often used in auction protocols, for example) which comes into play when it comes to force a party to choose a value without directly revealing it. Zero-knowledge sets (ZKS) [199] allow a prover to commit to a set of values S so as to be able to subsequently (and non-interactively) prove statements such as « element x belongs to the set S » or « element y does not belong to S » without revealing anything else, not even the overall cardinality of the set S. In collaboration with Moti Yung, we described [188] a ZKS protocol where proofs of membership and non-membership can both be short (less than 2 kB in implementations using suitable parameters). We thus improved upon previous ZKS schemes (and notably the construction of Catalano, Fiore and Messina [76]), where only proofs of non-membership can be made compact. So far, our construction remains the most efficient ZKS system in terms of communication complexity. In comparison with the first proposal of Micali, Rabin and Kilian [199], proofs are compressed to 13 % of their original length. In addition, we showed how to provide our scheme with certain non-malleability properties. Namely, we can prevent dishonest provers from correlating their hidden set to those of honest provers and still generating convincing proofs. In the same paper [188], as an intermediate result, we also proposed the first commitment scheme that allows committing to vectors of messages in such a way that the commitment – which has constant size – can be selectively opened with respect to one coordinate of the vector without revealing the content of other coordinates and with an opening of constant size (here, "constant" means independent of the dimension of the vector). As a second contribution to the area of commitment schemes, in collaboration with Marc Fischlin and Mark Manulis, we described [108] new constructions of universally composable commitments [71]. These are commitment schemes that, as required by Canetti's universal composition framework [70], provably remain secure in arbitrary environments, when composed with any other protocol. Universally composable (UC) commitments provide very strong security guarantees, including non-malleability, but they are notoriously very hard to construct (some setup assumption, like a common reference string generated by some trusted party, is inevitable, as shown by Canetti and Fischlin [71]). Yet, our new constructions feature a previously unique combination of efficiency and security properties. Namely, they are the first adaptively secure UC commitments where: (1) The sender can commit to multiple bits at once (so that n-bit strings can be committed to using $O(k+n)$ bits instead of $O(kn)$, where k is the security parameter); (2) The common reference string can be re-used across multiple commitments (and not only once as in certain constructions); (3) The commitment and opening phases both consist of a single message from the sender to the receiver.

0.7 Homomorphic Cryptography

Homomorphic signatures were first suggested by Desmedt [97] and formally defined by Johnson *et al.* [148]. They can be seen as the signature counterpart of homomorphic public-key encryption in that they allow a signer to authenticate messages in such a way that anyone can publicly derive a signature on certain functions of previously signed messages. In linearly homomorphic signatures [48], for example, the signer can authenticate vectors using his private key. Later on, anyone will be able to compute a signature on any linear combination of the signed vectors. As another example, homomorphic subset signatures [148, 11] make it possible for the signer to sign a set of values so that it will be possible to publicly derive a signature on a subset of the original set. Homomorphic signatures notably find applications in proofs of storage [13, 16] or proofs of correct computation [47, 46, 11] in cloud computing systems: when a client wants to outsource large datasets on a remote storage server, he can ask the latter to perform computations on his data. If the original dataset is signed by the client using a homomorphic signature scheme, the server will be able to authenticate the result of his computation, by publicly deriving a signature on the result of the carried out operation. For example, a linearly homomorphic scheme allows one to authenticate sums, averages or Fourier transforms on outsourced data: by verifying the signature derived by the server, the client will be convinced that the server properly archived his dataset and correctly computed the requested statistics. Certain applications need homomorphic signatures that satisfy certain privacy properties requiring derived signatures to be perfectly indistinguishable from original signatures. In proofs of correct computation, one may want the derived signature to hide all partial information about the original dataset: only the mean or the average should become public. If homomorphic subset signatures are used by an administration to authenticate e-ID cards, the latter privacy notion guarantees that the card holder will be able to prove that he is above 18 years old (by deriving a signature on the "date of birth" field of his ID card) without revealing his exact place of birth or any other private information. In collaboration with Nuttapong Attrapadung and Thomas Peters, we suggested stronger definitions of information-theoretic privacy for homomorphic signatures. In [25, 26], we also described the first constructions of homomorphic subset signatures and linearly homomorphic signatures that satisfy our strongest privacy notion in the standard model. We also described the most efficient (notably in terms of signature size) linearly homomorphic signature with a security proof under standard assumptions in the standard model. At PKC 2013, we also designed a homomorphic quotable signature scheme – where a signature on a string allows publicly computing a signature on any substring of the original string – satisfying the strongest privacy property while retaining signatures of optimal size.

In 2013, in collaboration with Marc Joye, Moti Yung and Thomas Peters [177], we showed a somewhat surprising application of linearly homomorphic signatures in the construction of non-interactive non-malleable commitments [101, 102] in the common reference string model. The goal of non-malleable commitments is to enforce the independence among distinct parties' committed values. To our knowledge, there was previously no efficient construction of non-interactive non-malleable commitment where a short commitment string allows committing to a vector while remaining able to efficiently prove properties about committed coordinates (which precludes the trivial solution consisting in committing to hashed vectors). In [177], we showed that any linearly homomorphic signature that fits a certain template – as is the case of all known constructions based on bilinear maps – can be turned into a primitive called non-interactive simulation-sound trapdoor commitment [116, 195] which, in turn, implies non-interactive non-malleable commitments in the sense of a definition used by Damgård and Groth [94]. Our construction yields constant-size commitments to vectors which preserve the ability to prove statements about committed vectors in a zero-knowledge manner (using interaction or not). In the same paper [177], we also considered linearly homomorphic signature schemes that are also structure-preserving. Namely, they make it possible to sign vectors of group elements of unknown discrete logarithms. We described efficient constructions of linearly homomorphic structure-preserving signatures (LHSPS) and used them to generically build non-malleable commitments to group elements. These were the first examples of non-malleable commitments allowing to prove knowledge of an opening using the Groth-Sahai techniques [138]. Later on [178], we also used linearly homomorphic structure-preserving signatures to build quasi-adaptive non-interactive zero-knowledge (QA-NIZK) proof systems, as defined by Jutla and Roy [151], with constant-size proofs. Specifically, our construction [178] allows proving that a vector of group elements $\mathbf{v} \in \mathbb{G}^n$ belongs to a linear subspace spanned by $t < n$ independent vectors of group elements $\mathbf{v}_1, \ldots, \mathbf{v}_t \in \mathbb{G}^n$. The novelty of our proof system – which is actually an argument system since only polynomially bounded adversaries are unable to prove false statements – is to provide constant-size proofs (typically made of 2 or 3 group elements), regardless of the dimension of the subspace. In addition, we showed how our QA-NIZK proof system can be endowed with a property called simulation-soundness [231], which basically prevents a probabilistic polynomial-time (PPT) adversary from proving false statements, even after having seen simulated proofs for possibly false statements. As an application, we described [178] more efficient non-interactive threshold cryptosystems that are both chosen-ciphertext-secure and secure against adaptive corruptions.

0.8 Organization

In the upcoming chapters, this thesis will give an overview of my results on the applications of structure-preserving cryptography. Chapter 1 will provide some background material which will ease the reading of subsequent chapters. Chapter 2 will describe my results on the design of group encryption [81] and revocable group signatures [180, 179], which are amongst my most important results on privacy-enhancing cryptographic protocols based on structure-preserving cryptography. Chapter 3 will finally present my constructions [178] of structure-preserving signatures with additive homomorphic properties and explain their applications in the design of non-interactive non-malleable primitives. These include non-malleable commitments, space-efficient simulation-sound QA-NIZK argument systems and chosen-ciphertext-secure public-key encryption.

List of Publications

Articles marked with [⋆] are the articles presented in this manuscript.
The articles below can be downloaded at `http://perso.ens-lyon.fr/benoit.libert/`.

Refereed Journals

[J1] Benoît Libert, Jean-Jacques Quisquater and Moti Yung. *Key Evolution Systems in Untrusted Update Environments* , extended version of [15], in *ACM Transactions on Information and System Security (ACM-TISSEC)*, December 2010, volume 13(4), Article 37.

[J2] Benoît Libert and Damien Vergnaud. *Unidirectional Chosen-Ciphertext Secure Proxy Re-Encryption*, extended version of [17], in IEEE Transactions on Information Theory, March 2011, volume 57(3), pp. 1786–1802.

[J3] Benoît Libert and Moti Yung. *Efficient Traceable Signatures in the Standard Model* , extended version of [22], in Theoretical Computer Science, March 2011, volume 412(12-14), pp. 1220–1242.

[J4] Benoît Libert and Damien Vergnaud. *Towards Practical Black-Box Accountable Authority IBE*: *Weak Black-Box Traceability with Short Ciphertexts and Private Keys* , extended version of [20], in IEEE Transactions on Information Theory, October 2011, volume 57(10), pp. 7189-7204.

[J5] Nuttapong Attrapadung and Benoît Libert. *Functional Encryption for Public-Attribute Inner Product*: *Achieving Constant-Size Ciphertexts with Adaptive Security or Support for Negation*, extended version of [27], in Journal of Mathematical Cryptology, October 2011, vol. 5(2), pp. 115-158.

[J6] Nuttapong Attrapadung, Javier Herranz, Fabien Laguillaumie, Benoît Libert, Elie De Panafieu and Carla Ràfols. *Attribute-Based Encryption Schemes with Constant-Size Ciphertexts*. Includes an extended version of [30], in Theoretical Computer Science, March 2012, vol. 422, pp. 15-38, 2012.

[J7] Benoît Libert and Moti Yung. *Adaptively Secure Non-Interactive Threshold Cryptosystems*, extended version of [32], in Theoretical Computer Science, March 2013, Vol. 478, pp. 76–100.

Papers in international conferences with scientific committee and proceedings

[1] Benoît Libert & Jean-Jacques Quisquater. *New identity based signcryption schemes from pairings*, in *IEEE Information Theory Workshop (ITW) 2003*, (J. Boutros ed.), IEEE, 2003, p. 155-158.

[2] Benoît Libert & Jean-Jacques Quisquater. *Efficient Revocation and Threshold Pairing Based Cryptosystems*, in *22nd Symposium on Principles of Distributed Computing (PODC 2003)*, (S. Rajsbaum ed.), ACM Press, 2003, p. 163-171.

[3] Benoît Libert & Jean-Jacques Quisquater. *Identity Based Undeniable Signatures*, in *Topics in Cryptology -* CT-RSA *2004* (T. Okamoto, ed.), Lect. Notes Comput. Sci., vol. 2964, Springer, 2004, p. 112-125.

[4] Benoît Libert & Jean-Jacques Quisquater. *Efficient Signcryption with Key Privacy from Gap Diffie-Hellman Groups*, in *Public Key Cryptography (*PKC*) 2004* (F. Bao, ed.), Lect. Notes Comput. Sci., vol. 2947, Springer, 2004, p. 187-200.

[5] Julien Cathalo, Benoît Libert & Jean-Jacques Quisquater. *Cryptanalysis of a Verifiably Committed Signature Scheme based on GPS and RSA*, in *Information Security Conference (*ISC*) 2004* (K. Zhang & Y. Zheng, ed.), Lect. Notes Comput. Sci., vol. 3225, Springer, 2004, p. 52–60.

[6] Benoît Libert & Jean-Jacques Quisquater. *Improved Signcryption from q-Diffie-Hellman Problems*, in *Fourth Conference on Security in Communication Networks, SCN 2004* (C. Blundo & S. Cimato, eds.), Lect. Notes Comput. Sci., vol. 3352, Springer, 2005, p. 220–234.

[7] Benoît Libert & Jean-Jacques Quisquater. *Identity Based Encryption without Redundancy*, in *Applied Cryptography and Network Security (*ACNS*) 2005* (J. Ioannidis, A. Keromytis & M. Yung eds.), Lect. Notes Comput. Sci., vol. 3531, Springer, 2005, p. 285-300.

[8] Paulo Barreto, Benoît Libert, Noel McCullagh & Jean-Jacques Quisquater. *Efficient and Provably-Secure Identity-Based Signatures and Signcryption from Bilinear Maps*, in *Advances in Cryptology -* ASIACRYPT *2005*, (B. Roy ed.), Lect. Notes Comput. Sci., vol. 3788, Springer, 2005, p. 515-532.

[9] Julien Cathalo, Benoît Libert & Jean-Jacques Quisquater. *Efficient and Non-interactive Timed-Release Encryption*, in *Information and Communications Security, 7th International Conference, ICICS 2005* (J. Lopez, W. Mao, S. Qing & G. Wang eds.), Lect. Notes Comput. Sci., vol. 3783, Springer, 2004, p. 291-303.

[10] Benoît Libert & Jean-Jacques Quisquater. *On Constructing Certificateless Cryptosystems from Identity Based Encryption*, in *Public Key Cryptography (PKC) 2006* (M. Yung ed.), Lect. Notes Comput. Sci., vol. 3958, Springer, 2007, p. 474-490.

[11] Fabien Laguillaumie, Benoît Libert & Jean-Jacques Quisquater. *Universal Designated Verifier Signatures Without Random Oracles or Non-Black Box Assumptions*, in *Security and Cryptography for Networks (SCN) 2006*, (R. De Prisco & M. Yung eds.), Lect. Notes Comput. Sci., vol. 4116, Springer, 2007, p. 63–77.

[12] Benoît Libert, Jean-Jacques Quisquater & Moti Yung. *Efficient Intrusion-Resilient Signatures Without Random Oracles*, in *2nd International Conference on Information Security and Cryptology (Inscrypt 2006)*, Lect. Notes Comput. Sci., vol. 4318, Springer, 2006, p. 27–41.

[13] Benoît Libert, Jean-Jacques Quisquater & Moti Yung. *Parallel Key-Insulated Public Key Encryption Without Random Oracles*, in *Public Key Cryptography (PKC) 2007* (T. Okamoto & X. Wang eds.), Lect. Notes Comput. Sci., vol. 4450, Springer, 2007, p. 298–314.

[14] Benoît Libert & Jean-Jacques Quisquater. *Practical Time Capsule Signatures in the Standard Model from Bilinear Maps*, in *1st International Conference on Pairing-based Cryptography* – PAIRING *2007*, (T. Takagi & T. Okamoto eds.), Lect. Notes Comput. Sci., vol. 4575, Springer, 2007, p. 23-38.

[15] Benoît Libert, Jean-Jacques Quisquater & Moti Yung. *Forward-secure signatures in untrusted update environments: efficient and generic constructions*, in *14th ACM Conference on Computer and Communications Security (ACM-CCS) 2007* (S. De Capitani di Vimercati & P. Syverson eds.), ACM Press, 2007, p. 266–275.

[16] Alexander W. Dent, Benoît Libert & Kenneth G. Paterson. *Certificateless Encryption Schemes Strongly Secure in the Standard Model*, in *Public Key Cryptography (PKC) 2008* (R. Cramer ed.), Lect. Notes Comput. Sci., vol. 4939, Springer, 2008, p. 344-359.

[17] Benoît Libert & Damien Vergnaud. *Unidirectional Chosen-Ciphertext Secure Proxy Re-Encryption*, in *Public Key Cryptography (PKC) 2008* (R. Cramer ed.), Lect. Notes Comput. Sci., vol. 4939, Springer, 2008, p. 360-379.

[18] Benoît Libert & Damien Vergnaud. *Tracing Malicious Proxies in Proxy Re-Encryption*, in *2nd International Conference on Pairing-Based Cryptography (Pairing 2008)*, (S. Galbraith & K. Paterson eds.), Lect. Notes Comput. Sci., vol. 5209, Springer, 2008, p. 332-353.

[19] Benoît Libert & Damien Vergnaud. *Multi-Use Unidirectional Proxy Re-Signatures*, in *15th ACM Conference on Computer and Communications Security (ACM-CCS) 2008* (P. Syverson & S. Jha eds.), ACM Press, 2008, p. 511-520.

[20] Benoît Libert & Damien Vergnaud. *Towards Black-Box Accountable Authority IBE with Short Ciphertexts and Private Keys*, in *Public Key Cryptography (PKC) 2009*, (G. Tsudik & S. Jarecki eds.), Lect. Notes Comput. Sci., vol. 5443, Springer, 2009, p. 235-255.

[21] Benoît Libert & Damien Vergnaud. *Adaptive-ID Secure Revocable Identity-Based Encryption*, in *Topics in Cryptology* - CT-RSA *2009*, (M. Fischlin ed.), Lect. Notes Comput. Sci., vol. 5473, Springer, 2008, p. 1–15.

[22] Benoît Libert & Moti Yung. *Efficient Traceable Signatures in the Standard Model*, in *3rd International Conference on Pairing-Based Cryptography* - PAIRING *2009* (H. Shacham & B. Waters eds.), Lect. Notes Comput. Sci., vol. 5671, Springer, 2009, p. 187–205.

*[23] Julien Cathalo, Benoît Libert & Moti Yung. *Group Encryption*: *Non-Interactive Realization in the Standard Model*, in *Advances in Cryptology - ASIACRYPT 2009* (M. Matsui ed.), Lect. Notes Comput. Sci., vol. 5912, Springer, 2009, p. 179–196.

[24] Benoît Libert & Damien Vergnaud. *Group Signatures with Verifier-Local Revocation and Backward Unlinkability in the Standard Model*, in *8th International Conference on Cryptology and Network Security (CANS 2009)*, (J. Garay & A . Miyaji eds.), Lect. Notes Comput. Sci., vol. 5888, Springer, 2009, p. 498–517.

[25] Benoît Libert & Moti Yung. *Concise Mercurial Vector Commitments and Independent Zero-Knowledge Sets with Short Proofs*, in *7th Theory of Cryptography Conference -* TCC *2010* (D. Micciancio ed.), Lect. Notes Comput. Sci., vol. 5978, Springer, 2010, p. 499–517.

[26] Benoît Libert & Moti Yung. *Dynamic Fully Forward-Secure Group Signatures*, in *5th ACM Symposium on Information, Computer and Communications Security (AsiaCCS) 2010* (D. Basin ed.), ACM Press, 2010, p. 70–81.

[27] Nuttapong Attrapadung & Benoît Libert. *Functional Encryption for Inner Product*: *Achieving Constant-Size Ciphertexts with Adaptive Security or Support for Negation*, in *Public Key Cryptography (PKC) 2010* (P. Nguyen & D. Pointcheval eds.), Lect. Notes Comput. Sci., vol. 6056, Springer, 2010, p. 384–402.

[28] David Galindo, Benoît Libert, Marc Fischlin, Georg Fuchsbauer, Anja Lehmann, Mark Manulis & Dominique Schröder. *Public-Key Encryption with Non-Interactive Opening*: *New Constructions and Stronger Definitions*, in *Africacrypt 2010* (D. Bernstein & T. Lange eds.), Lect. Notes Comput. Sci., vol. 6055, Springer, 2010, p. 333–350.

[29] Benoît Libert & Moti Yung. *Efficient Completely Non-Malleable Public Key Encryption*, in *37th International Colloquium on Automata, Languages and Programming (ICALP) 2010 - Track A (Algorithms, Complexity and Games)* (P. Spirakis ed.), Lect. Notes Comput. Sci., vol. 6198 , Springer, 2010, p. 127–139.

[30] Nuttapong Attrapadung & Benoît Libert. *Homomorphic Network Coding Signatures in the Standard Model*, in *Public Key Cryptography (PKC) 2011* (D. Catalano, N. Fazio, R. Gennaro & A. Nicolosi eds.), Lect. Notes Comput. Sci., vol. 6571, Springer, 2011, p. 17–34.

[31] Nuttapong Attrapadung, Benoît Libert & Elie de Panafieu. *Expressive Key-Policy Attribute-Based Encryption with Constant-Size Ciphertexts*, in *Public Key Cryptography (PKC) 2011*, (D. Catalano, N. Fazio, R. Gennaro & A. Nicolosi eds.), Lect. Notes Comput. Sci., vol. 6571, Springer, 2011, p. 90–108.

[32] Benoît Libert & Moti Yung. *Adaptively Secure Non-Interactive Threshold Cryptosystems*, in *38th International Colloquium on Automata, Languages and Programming (ICALP) 2011 - Track C (Models, Algorithms and Information Management)* (M. Henzinger, L. Aceto & J. Sgall eds.), Lect. Notes Comput. Sci., vol. 6756, Springer, p. 588–600, 2011.

[33] Brett Hemenway, Benoît Libert, Rafail Ostrovsky & Damien Vergnaud. *Lossy Encryption: Constructions from General Assumptions and Efficient Selective Opening Chosen Ciphertext Security*, in *Advances in Cryptology* - ASIACRYPT *2011* (D.-H. Lee & X. Wang eds.), Lect. Notes Comput. Sci., vol. 7073, Springer, p. 70–88, 2011.

[34] Marc Fischlin, Benoît Libert & Mark Manulis. *Non-Interactive and Re-Usable Universally Composable String Commitments with Adaptive Security*, in *Advances in Cryptology* - ASIACRYPT *2011* (D.-H. Lee & X. Wang eds.), Lect. Notes Comput. Sci., vol. 7073, p. 468–485, Springer, 2011.

[35] Malika Izabachène, Benoît Libert & Damien Vergnaud. *Block-Wise P-Signatures and Non-Interactive Anonymous Credentials with Efficient Attributes*, in *IMA International Conference on Cryptography and Coding* (IMACC) *2011* (L. Chen ed.), Lect. Notes Comput. Sci., vol. 7089, p. 431–450, Springer, 2011.

[36] Javier Herranz, Fabien Laguillaumie, Benoît Libert & Carla Ràfols. *Short Attribute-Based Signatures for Threshold Predicates*, in *Topics in Cryptology* - CT-RSA *2012* (O. Dunkelman ed.), Lect. Notes Comput. Sci., vol. 7178, p. 51–67, Springer, 2012.

[37] Benoît Libert & Moti Yung. *Non-Interactive CCA-Secure Threshold Cryptosystems with Adaptive Security: New Framework and Constructions*, in *9th Theory of Cryptography Conference (TCC 2012)* (R. Cramer ed.), Lect. Notes Comput. Sci., vol. 7194, p. 75–93, Springer, 2012.

⋆[38] Benoît Libert, Thomas Peters & Moti Yung. *Scalable Group Signatures with Revocation*, in *Advances in Cryptology* - EUROCRYPT *2012* (D. Pointcheval & T. Johansson eds.), Lect. Notes Comput. Sci., vol. 7237, p. 609–627, Springer, 2012.

[39] Benoît Libert, Kenneth G. Paterson & Elizabeth A. Quaglia. *Anonymous Broadcast Encryption: Adaptive Security and Efficient Constructions in the Standard Model*, in *Public Key Cryptography (PKC) 2012* (M. Fischlin, J. Buchmann & M. Manulis eds.), Lect. Notes Comput. Sci., vol. 7293, p. 206–224, Springer, 2012.

[40] Malika Izabachène & Benoît Libert. *Divisible E-Cash in the Standard Model*, in *5th International Conference on Pairing-Based Cryptography* - PAIRING *2012* (M. Abdalla & T. Lange eds.), Lect. Notes Comput. Sci. vol. 7708, p. 314–332, Springer, 2012.

⋆[41] Benoît Libert, Thomas Peters & Moti Yung. *Group Signatures with Almost-for-free Revocation*, in *Advances in Cryptology* - CRYPTO *2012* (R. Safavi-Naini & R. Canetti eds.), Lect. Notes Comput. Sci. vol. 7417, p. 571-589, Springer, 2012.

[42] Nuttapong Attrapadung, Benoît Libert & Thomas Peters. *Computing on Authenticated Data: New Privacy Definitions and Constructions*, in *Advances in Cryptology* - ASIACRYPT *2012* (X. Wang & K. Sako eds.), Lect. Notes Comput. Sci. vol. 7658, p. 367-385, Springer, 2012.

[43] Pooya Farshim, Benoît Libert, Kenneth G. Paterson & Elizabeth Quaglia. *Robust Encryption, Revisited*, in PUBLIC KEY CRYPTOGRAPHY *(PKC) 2013* (K. Kurosawa ed.), Lect. Notes Comput. Sci. vol. 7778, p. 352-368, Springer, 2013.

[44] Nuttapong Attrapadung, Benoît Libert & Thomas Peters. *Efficient Completely Context Hiding Quotable and Linearly Homomorphic Signatures*, in PUBLIC KEY CRYPTOGRAPHY *(PKC) 2013*, (K. Kurosawa ed.), Lect. Notes Comput. Sci. vol. 7778, p. 386-404, Springer, 2013.

[45] Marc Joye & Benoît Libert. *A Scalable Scheme for Privacy-Preserving Aggregation of Time-Series Data*, in FINANCIAL CRYPTOGRAPHY AND DATA SECURITY *(FC) 2013* (A. Sadeghi ed.), Lect. Notes Comput. Sci. vol. 7859, p. 111-125, Springer, 2013.

[46] Marc Joye & Benoît Libert. *Efficient Cryptosystems from* 2^k*-th Power Residue Symbols*, in *Advances in Cryptology* - EUROCRYPT *2013* (T. Johansson & P. Nguyen eds.), Lect. Notes Comput. Sci. vol. 7881, p. 76-92, Springer, 2013.

⋆[47] Benoît Libert, Thomas Peters, Marc Joye & Moti Yung. *Linearly Homomorphic Structure-Preserving Signatures and their Applications*, in *Advances in Cryptology* - CRYPTO *2013* (R. Canetti & J. Garay eds.), Lect. Notes Comput. Sci. vol. 8043, p. 289-307, Springer, 2013.

[48] Fabien Laguillaumie, Adeline Langlois, Benoît Libert & Damien Stehlé. *Lattice-Based Group Signatures with Logarithmic Signature Size*, in *Advances in Cryptology* - ASIACRYPT *2013* (K. Sako & P. Sarkar eds.), Lect. Notes Comput. Sci. vol. 8270, p. 41-61, Springer, 2013.

[49] Benoît Libert & Marc Joye. *Group Signatures with Message-Dependent Opening in the Standard Model*, in *Topics in Cryptology* - CT-RSA *2014* (J. Benaloh ed.), Lect. Notes Comput. Sci. vol. 8366, p. 286-306, Springer, 2014.

[50] Alex Escala, Javier Herranz, Benoît Libert & Carla Ràfols. *Identity-Based Lossy Trapdoor Functions*: *New Definition, Hierarchical Extensions, and Implications*, in *Public Key Cryptography (*PKC*) 2014* (H. Krawczyk ed.), Lect. Notes Comput. Sci. vol. 8383, p. 239-256, Springer, 2014.

[51] Benoît Libert, Moti Yung, Marc Joye & Thomas Peters. *Traceable Group Encryption*, in *Public Key Cryptography (*PKC*) 2014* (H. Krawczyk ed.), Lect. Notes Comput. Sci. vol. 8383, p. 592-610, Springer, 2014.

⋆[52] Benoît Libert, Thomas Peters, Marc Joye & Moti Yung. *Non-Malleability from Malleability*: *Simulation-Sound Quasi-Adaptive NIZK Proofs and CCA2-Secure Encryption from Homomorphic Signatures*, in *Advances in Cryptology* - EUROCRYPT *2014* (P. Nguyen & E. Oswald eds.), Lect. Notes Comput. Sci. vol. 8441, p. 514-532, Springer, 2014.

[53] Benoît Libert, Marc Joye & Moti Yung. *Born and Raised Distributively*: *Fully Distributed Non-Interactive Adaptively-Secure Threshold Signatures with Short Shares*, in *33rd Symposium on Principles of Distributed Computing (PODC 2014)*, (S. Dolev ed.), p. 303-312, ACM Press, 2014.

[54] Benoît Libert, Marc Joye, Moti Yung and Thomas Peters. *Concise Multi-Challenge CCA-Secure Encryption and Signatures with Almost Tight Security*. In *Advances in Cryptology* - ASIACRYPT *2014* (P. Sarkar & T. Iwata eds.), Lect. Notes Comput. Sci. series, Springer, 2014.

CHAPTER 1

Background

This chapter briefly recalls several notions and definitions that are related to non-interactive zero-knowledge proofs and structure-preserving cryptography. These reminders will make it easier to explain the results of subsequent chapters.

1.1 Bilinear Maps and Hardness Assumptions

Definition 1 (Bilinear Groups). *A bilinear group system is a tuple $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$ where $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are cyclic abelian groups of prime order $p > 2^\lambda$, where $\lambda \in \mathbb{N}$ is a security parameter, generated respectively by $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$ and $e(g_1, g_2) \in \mathbb{G}_T$. If $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a non-degenerated bilinear form, for all $X \in \mathbb{G}_1$, for all $Y \in \mathbb{G}_2$, for all $a, b \in \mathbb{Z}_p$,*

$$e(X^a, Y^b) = e(X, Y)^{ab}. \tag{1.1}$$

For a security parameter λ, it is assumed that bilinear groups are efficiently samplable so that $p > 2^\lambda$. Mainly, there are three types of elliptic-curve instantiations [115]:

Type I: *where $\mathbb{G}_1 = \mathbb{G}_2$ and $g_1 = g_2$. We usually refer to Type-I instances as* symmetric pairings*. We denote by $(p, \mathbb{G}, \mathbb{G}_T, e, g) \leftarrow \Lambda(\lambda)$ the generation of this setting.*

Type II: *where $\mathbb{G}_1 \neq \mathbb{G}_2$ and an efficient isomorphism $\psi : \mathbb{G}_2 \to \mathbb{G}_1$ is available but none is efficiently computable from $\mathbb{G}_1$ to $\mathbb{G}_2$.*

Type III: *where $\mathbb{G}_1 \neq \mathbb{G}_2$ but no efficient isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$ is efficiently computable in either direction.*

Type III elliptic curves have the most efficient instantiations and admit a smaller representation of $\mathbb{G}_1$-elements than those of $\mathbb{G}_2$-elements. At a same bit-security level, $\mathbb{G}$-elements of Type I elliptic curves have an intermediate size relatively to Type III curves. In the following chapters we will often use Type I groups in order to keep the description of systems as simple as possible. We will, however, mention extensions to Type II or Type III pairings whenever they are possible.

1.1.1 Algorithmic Assumptions

All the schemes proposed in the thesis have their security based on one or several of the following assumptions. To simplify their descriptions we will say "a problem is hard in a group $\mathbb{G}$" for "a problem is hard relatively to the generation of $\mathbb{G}$", which means that the probability to efficiently solve the problem is negligible in the security parameter λ where the random coins are taken over the distribution of the λ-bit length instance of the problem and the distribution that generates the group $\mathbb{G}$ whose cardinality is at least 2^{λ}. In symmetric bilinear groups, the latter distribution is that of $\Lambda(\lambda)$ such that $(p, \mathbb{G}, \mathbb{G}_T, e, g) \leftarrow \Lambda(\lambda)$.

As a warm-up, we start with the weakest assumption of the thesis. Breaking this assumption means breaking all the other ones since the underlying problem is the hardest to solve. For a set S, $s \overset{\$}{\leftarrow} S$ means that s is equally-likely sampled from S.

Assumption 1 (DLOG)**.** *The* **Discrete Logarithm** *(DLOG) problem in a cyclic group $(p, \mathbb{G}, g)$, is to compute $a \in \mathbb{Z}_p$ such that $h = g^a$ for some $h \overset{\$}{\leftarrow} \mathbb{G}$. The* **Discrete Logarithm Assumption** *asserts that the DLOG problem is hard in $\mathbb{G}$.*

Assumption 2 (CDH)**.** *In a cyclic group $\mathbb{G} = \langle g \rangle$ of order p, the* **Computational Diffie-Hellman** *(CDH) problem is, given $(g, g^a, g^b) \in \mathbb{G}^3$, for some $a, b \overset{\$}{\leftarrow} \mathbb{Z}_p^*$, to compute $g^{ab} \in \mathbb{G}$. The* **Computational Diffie-Hellman Assumption** *posits the intractability of the CDH problem in the group $\mathbb{G}$.*

In some cases, reductions from the hardness of CDH may be difficult to obtain. In such situations, the following assumption is sometimes convenient to use.

Assumption 3 (Flex-CDH [163])**.** *The* ***Flexible Diffie-Hellman Assumption*** *(Flex-CDH) in $\mathbb{G}$ asserts the hardness of finding a non-trivial triple $(g^{\mu}, g^{a \cdot \mu}, g^{ab \cdot \mu}) \in (\mathbb{G} \backslash \{1_{\mathbb{G}}\})^3$, for some non-zero $\mu \in \mathbb{Z}_p^*$, given $(g, g^a, g^b) \overset{\$}{\leftarrow} \mathbb{G}$.*

When it comes to proving indistinguishability-based security, the hardness of decisional problems often come in handy. A well-known decisional assumption is the difficulty of the Decision Diffie-Hellman DDH problem which amounts to recognizing the solution of a CDH instance.

Assumption 4 (DDH)**.** *In a cyclic group $\mathbb{G} = \langle g \rangle$ of order p, the* **Decision Diffie-Hellman** *(DDH) problem, is to distinguish the distributions (g, g^a, g^b, g^{ab}) and (g, g^a, g^b, g^c), with $a, b \overset{\$}{\leftarrow} \mathbb{Z}_p$, $c \overset{\$}{\leftarrow} \mathbb{Z}_p$. The* **Decision Diffie-Hellman Assumption** *posits that DDH is hard in $\mathbb{G}$. The DDH assumption holds in $\mathbb{G}$ if, for any PPT distinguisher $\mathcal{A}$, it holds that*

$$\mathbf{Adv}_{\mathcal{A}}^{\mathrm{DDH}}(\lambda) = \Big| \Pr[\mathcal{A}(g, g^a, g^b, g^{ab}) = 1 \mid a, b \overset{R}{\leftarrow} \mathbb{Z}_p] \\ - \Pr[\mathcal{A}(g, g^a, g^b, g^c) = 1 \mid a, b, c \overset{R}{\leftarrow} \mathbb{Z}_p] \Big| \in \mathsf{negl}(\lambda),$$

where the probabilities are taken over all coin tosses.

In symmetric bilinear groups $(p, \mathbb{G}, \mathbb{G}_T, e, g) \leftarrow \Lambda(\lambda)$, the DDH assumption does not hold. Indeed, given $(g, h, f, T) \in \mathbb{G}^4$, deciding whether $T = f^{\log_g(h)}$ can be done efficient by checking whether $e(g, T) = e(h, f)$.

On the other hand, the DDH assumption is believed [234] to hold in $\mathbb{G}_1$ for asymmetric bilinear groups $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$ of Type II since there is no apparent way to invert the isomorphism $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$. In Type III configurations $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$ (where no isomorphism is efficiently computable in either direction between $\mathbb{G}_1$ and $\mathbb{G}_2$), the DDH assumption is believed to hold in both $\mathbb{G}_1$ and $\mathbb{G}_2$. The simultaneous intractability of DDH in $\mathbb{G}_1$ and $\mathbb{G}_2$ for Type III pairings is called Symmetric eXternal Diffie-Hellman assumption (SXDH) [234].

In symmetric pairings, the hardness of the DLIN problem appears as a reasonable assumption to rely on.

Assumption 5 (DLIN [44]). *In a cyclic group $\mathbb{G} = \langle g \rangle$ of order p, the* **Decision Linear** *(DLIN) problem is to distinguish the distributions $(g^a, g^b, g^{ac}, g^{bd}, g^{c+d})$ and $(g^a, g^b, g^{ac}, g^{bd}, g^z)$, with $a, b, c, d \xleftarrow{\$} \mathbb{Z}_p$, $z \xleftarrow{\$} \mathbb{Z}_p$. The* **Decision Linear Assumption** *is the intractability of DLIN for any PPT distinguisher $\mathcal{D}$. The advantage of a distinguisher is defined analogously to the DDH case.*

Equivalently, for random group elements $g, h, f \leftarrow \mathbb{G}^3$, the DLIN assumption is the hardness of deciding whether an given triple $(f^c, h^d, Z) \in \mathbb{G}^3$, for unknown $(c, d) \in \mathbb{Z}_p^2$, satisfies $(f^c, h^d, Z) \in \mathsf{span}\langle (f, 1, g), (1, h, g) \rangle$ (i.e., $Z = g^{c+d}$), where span stands for the linear span of two or more vectors.

Assumption 6 (DP [4]). *In asymmetric bilinear groups $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$, the* ***Double Pairing*** *(DP) problem is, given $g_z, g_r \xleftarrow{\$} \mathbb{G}_1$, to find a non-trivial $(z, r) \in (\mathbb{G}_2 \backslash \{1_{\mathbb{G}_2}\})^2$ satisfying $1_{\mathbb{G}_T} = e(g_z, z) \cdot e(g_r, r)$. The* **Double Pairing Assumption** *asserts that the DBP problem is hard in $\mathbb{G}$.*

It is easy to see that the DP assumption is implied by the DDH assumption in $\mathbb{G}_1$. Given a DDH instance $(g_z, g_r, g_z^{\theta}, g_r^{\theta'})$, for any non-trivial pair $(z, r) \in \mathbb{G}_2^2$ satisfying the equality $e(g_z, z) \cdot e(g_r, r) = 1_{\mathbb{G}_T}$, we have $\theta = \theta'$ if and only if $e(g_z^{\theta}, z) \cdot e(g_r^{\theta'}, r) = 1_{\mathbb{G}_T}$.

In symmetric pairings, the DP and DDH problems are both easy. However, the DP assumption has an analogue, which we introduced in [81], that seems to hold in Type I pairings. This assumption is called Simultaneous Double Pairing (SDP) and, as shown in [81], it is implied by DLIN.

Assumption 7 (SDP [81]). *The* ***Simultaneous Double Pairing Problem*** *(SDP) in a symmetric bilinear group $(p, \mathbb{G}, \mathbb{G}_T, e, g) \leftarrow \Lambda(\lambda)$ is, given $g_z, g_r, h_z, h_u \xleftarrow{\$} \mathbb{G}^4$, to find $(z, r, u) \in \mathbb{G}^3$ satisfying the equalities*

$$1_{\mathbb{G}_T} = e(g_z, z) \cdot e(g_r, r), \qquad 1_{\mathbb{G}_T} = e(h_z, z) \cdot e(h_u, u). \tag{1.2}$$

The **Simultaneous Double Pairing Assumption** *is the hardness of the SDP problem.*

The assumption can be generalized to asymmetric pairing configurations $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$. If g_z, g_r, h_z, h_u are in $\mathbb{G}_1$ (resp. $\mathbb{G}_2$), finding a non-trivial $(z, r, u) \in \mathbb{G}_2^3$ (resp. $(z, r, u) \in \mathbb{G}_1^3$) such that (resp. $e(z, g_z) \cdot e(r, g_r) = e(z, h_z) \cdot e(u, h_u) = 1_{\mathbb{G}_T}$) is at least as hard as breaking the DLIN assumption in $\mathbb{G}_1$ (resp. $\mathbb{G}_2$).

In the symmetric setting, the connection between SDP and DLIN was observed [81] by noticing that, given a DLIN instance $(g_r, h_u, g, g_r^{\theta_1}, h_u^{\theta_2}, T)$ where either $T = g^{\theta_1 + \theta_2}$ or $T \in_R \mathbb{G}$, for any triple $(z, r, u) \in \mathbb{G}$ such that $e(g_r^{\theta_1}, z) \cdot e(g_r, r) = e(h_u^{\theta_2}, z) \cdot e(h_u, u) = 1_{\mathbb{G}_T}$, we have the equivalence

$$T = g^{\theta_1 + \theta_2} \quad \Leftrightarrow \quad e(T, z) \cdot e(g, r \cdot u) = 1_{\mathbb{G}_T}.$$

The DLIN assumption can be generalized as the problem of deciding whether $K + 1$ vectors of dimension $K + 1$ are linearly independent.

Assumption 8 (K-LIN [235, 143]). *In a cyclic group $\mathbb{G} = \langle g \rangle$ of order p, the* **K-Linear** *(K-LIN) problem is to distinguish the distributions*

$$\{(g_1^{a_1}, g_2^{a_2}, \ldots, g_K^{a_K}, g^{\sum_{i=1}^{K} a_i}) \mid g_1, \ldots, g_K \xleftarrow{\$} \mathbb{G},\ a_1, \ldots, a_K \xleftarrow{\$} \mathbb{Z}_p\}$$

and

$$\{(g_1^{a_1}, g_2^{a_2}, \ldots, g_K^{a_K}, g^{z}) \mid g_1, \ldots, g_K \xleftarrow{\$} \mathbb{G},\ a_1, \ldots, a_K, z \xleftarrow{\$} \mathbb{Z}_p\}.$$

The K-linear assumption is the infeasibility of K-LIN for any PPT algorithm.

The DDH and DLIN assumptions can be seen as special cases of the K-LIN assumption for $K = 1$ and $K = 2$, respectively. The difficulty of the problem is believed to increase with the dimension K. In the generic group model, it was shown [235, 143] that, for each $K > 1$, the K-linear problem remains hard in the presence of an oracle solving $(K-1)$-linear instances.

The SDP assumption as a similar generalization, which is implied by the K-linear assumption in the same way as SDP is implied by DLIN.

Assumption 9. *The* **Simultaneous K-wise Pairing** *(K-SDP) problem is, given a random tuple*

$$(g_{1,z}, \ldots, g_{K,z}, g_{1,r}, \ldots, g_{K,r}) \in_R \mathbb{G}^{2K},$$

to find a non-trivial vector $(z, r_1, \ldots, r_K) \in \mathbb{G}^{K+1}$ such that

$$e(g_{j,z}, z) \cdot e(g_{j,r}, r_j) = 1_{\mathbb{G}_T} \qquad j \in \{1, \ldots, K\} \tag{1.3}$$

and $z \neq 1_{\mathbb{G}}$.

Given a K-linear instance $(g_{1,r}, \ldots, g_{k,r}, g_{1,r}^{a_1}, \ldots, g_{K,r}^{a_K}, \eta) \in \mathbb{G}^{2K+1}$, for any non-trivial tuple $(z, r_1, \ldots, r_K)$ satisfying $e(g_{j,r}^{a_j}, z) \cdot e(g_{j,r}, r_j) = 1_{\mathbb{G}_T}$ for each $j \in \{1, \ldots, k\}$, we have

$$\eta = g^{\sum_{j=1}^{K} a_j} \qquad \Leftrightarrow \qquad e(g, \prod_{j=1}^{K} r_j) \cdot e(z, \eta) = 1_{\mathbb{G}_T}.$$

Hence, any algorithm solving K-SDP with non-negligible probability implies a K-linear distinguisher.

All the above assumptions can be classified in the category of *simple* assumptions [249]. By "simple assumption", we mean an assumption which is simultaneously falsifiable[1] [206] and with a description made of a constant number of group elements. In particular, the number of input elements does not depend on the number of queries made by the adversary or any feature (such as the maximal number of users in a system) of a specific cryptographic scheme. Simple assumptions are usually deemed more reliable than so-called q-type assumptions, which are parametrized and variable-length assumptions.

In some applications, more efficient schemes may be obtained by relying on a family of q-type assumptions. While these assumptions are usually falsifiable, the number of group elements in a problem instance depends on a parameter q determined by the cryptographic system (e.g., the maximal number of members in a group of users) or the power of adversary (via the number of queries). The strength of the assumption thus depends on the desired scalability of the considered protocol or the resources made available to the adversary. However, the assumptions described in this section all resist generic adversaries [237].

[1] Namely, it should be possible to publicize a problem instance as a challenge and efficiently check the correctness of any candidate solution to this challenge.

q-type assumptions We also rely on assumptions that can be seen as non-interactive variants of "one-more" problems, where the goal of the problem solver is to find a new solution given q initial solutions. However, a difference between q-type problems and one-more problems is that, in in the former, the solver is given q inputs at once at the beginning instead of dynamically interacting with an oracle. Still, the length and the strength of the assumption are determined by a parameter q, which usually depends on the scalability of the system or the power of the adversary. For example, in the first use of the q-Strong Diffie-Hellman assumption [41], q was the number of signing queries made by the adversary.

Assumption 10 (q-SDH [41])**.** *The* **q-Strong Diffie-Hellman problem** *(q-SDH) in a group $(p, \mathbb{G}, g)$ is, given $(g, g^a, \ldots, g^{(a^q)})$, for some $a \stackrel{\$}{\leftarrow} \mathbb{Z}_p$, to find a pair $(g^{1/(a+s)}, s) \in \mathbb{G} \times \mathbb{Z}_p$. The* **$q$-Strong Diffie-Hellman Assumption** *asserts the hardness of the q-SDH problem.*

In [56], Boyen and Waters considered the following variant of the q-SDH assumption.

Assumption 11 ([56])**.** *The* **q-Hidden Strong Diffie-Hellman problem** *(q-HSDH) in $\mathbb{G}$ consists in, given $(g, \Omega = g^\omega, u) \stackrel{\$}{\leftarrow} \mathbb{G}^3$ and triples $\{(g^{1/(\omega+s_i)}, g^{c_i}, u^{c_i})\}_{i=1}^{q}$ with $c_1, \ldots, c_q \stackrel{\$}{\leftarrow} \mathbb{Z}_p$, finding another triple $(g^{1/(\omega+c)}, g^c, u^c)$ such that $c \neq c_i$ for $i = 1, \ldots, q$.*

While stronger than the q-SDH assumption, the q-HSDH assumption was shown [56] to hold in generic bilinear groups.

The following assumption has been used to prove the security of a constant-size structure-preserving signature [4, 6] scheme that allows signing vectors of group elements. It will also serve as a building block for some of our constructions in the forthcoming chapters.

Assumption 12 (q-SFP [6])**.** *The* **q-Simultaneous Flexible Pairing Problem** *(q-SFP) in a symmetric bilinear group $(p, \mathbb{G}, \mathbb{G}_T, e, g)$ is, given $g_z, h_z, g_r, h_r, a, \tilde{a}, b, \tilde{b} \in \mathbb{G}$ and $q \in \mathsf{poly}(\lambda)$ tuples $(z_j, r_j, s_j, t_j, u_j, v_j, w_j) \in \mathbb{G}^7$ such that*

$$\begin{aligned} e(a, \tilde{a}) &= e(g_z, z_j) \cdot e(g_r, r_j) \cdot e(s_j, t_j) \\ e(b, \tilde{b}) &= e(h_z, z_j) \cdot e(h_r, u_j) \cdot e(v_j, w_j), \end{aligned} \tag{1.4}$$

to find a new tuple $(z^\star, r^\star, s^\star, t^\star, u^\star, v^\star, w^\star) \in \mathbb{G}^7$ satisfying relation (1.4) and such that $z^\star \neq 1_\mathbb{G}$ and $z^\star \neq z_j$ for $j \in \{1, \ldots, q\}$. The **q-Simultaneous Flexible Pairing** *assumption states that the q-SFP problem is intractable in $\mathbb{G}$.*

Assumption 13 (q-DHE [49])**.** *The* **q-Diffie-Hellman Exponent Problem** *(q-DHE) in a cyclic group $(p, \mathbb{G}, g)$ is, given $(g, g_1, \ldots, g_q, g_{q+2}, \ldots, g_{2q}) \in \mathbb{G}^{2q}$ such that $g_i = g^{(\alpha^i)}$ for each i and where $\alpha \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$, to compute the missing element $g_{q+1} = g^{(\alpha^{q+1})}$. The hardness of the q-DHE problem is referred to as the* **q-Diffie-Hellman Exponent** *assumption in $\mathbb{G}$.*

The latter assumption and the q-SDH assumption are somewhat incomparable. On one hand, the q-DHE assumption is stronger as the adversary is given more input elements for the same parameter q. On the other hand, unlike the q-SDH problem, any instance of the q-DHE problem has only one possible answer.

As observed in [66], the q-DHE problem is not easier than the q-Bilinear Diffie-Hellman Exponent (q-BDHE) problem defined by Boneh, Gentry and Waters [49], which is to compute

$e(g,h)^{(\alpha^{q+1})}$ on input of the same values and the additional element $h \in \mathbb{G}$. The generic hardness of q-DHE thus follows from the generic security of the family of assumptions analyzed by Boneh, Boyen and Goh [43].

We also appeal to a stronger variant of Assumption 13, which was defined in [145], where its generic hardness was proved. While the Flex-CDH assumption relaxes the resolution of the CDH problem, the following assumption relaxes the q-DHE problem in a similar way.

Assumption 14 (q-Flex-DHE). *In a cyclic group $\mathbb{G} = \langle g \rangle$ of prime order p, the* **Flexible q-Diffie-Hellman Exponent** *(q-FlexDHE) problem is, given $(g, g_1, \ldots, g_q, g_{q+2}, \ldots, g_{2q}) \in \mathbb{G}^{2q}$ where $g_i = g^{(\alpha^i)}$ for each i and with $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$, to find a triple $(g^\mu, g_{q+1}^\mu, g_{2q}^\mu) \in (\mathbb{G}\backslash\{1_\mathbb{G}\})^3$, for some non-zero $\mu \in \mathbb{Z}_p^*$ and where $g_{q+1} = g^{(\alpha^{q+1})}$. The* **Flexible q-Diffie-Hellman Exponent** *assumption is the hardness of the q-FlexDHE problem for any PPT adversary.*

1.2 Non-Interactive Zero-Knowledge and Witness Indistinguishable Proofs

Zero-knowledge proofs [127, 126] allow a prover to convince a verifier that a given statement x belongs to some specific language $\mathcal{L}$ without revealing anything beyond the fact that $x \in \mathcal{L}$. In a proof system for an NP language, the prover uses an additional private input w, called the *witness*, which allows *efficiently* generating a convincing proof. This witness is generally hard-to-compute for the verifier since, otherwise, the latter could get convinced without any help from the prover.

1.2.1 Definition and Security Notions

Let $\mathcal{V}$ be a set whose elements are efficiently recognizable. A family of relations $\mathcal{R}$ defines a hard-to-invert NP language $\mathcal{L} \subsetneq \mathcal{V}$ if, for a security parameter λ, given the description of a relation $R \leftarrow \mathcal{R}(\lambda)$, there exists an efficient algorithm for sampling a pair (x, w), made of a statement x and a witness w, such that $R(x, w) = 1$. Moreover, given only the statement $x \in \mathcal{L} := \{x \in \mathcal{V} \,|\, \exists w : R(x, w) = 1\}$, it is computationally hard to compute a witness w such that $R(x, w) = 1$.

A language $\mathcal{L} \subset \mathcal{V}$ is said *hard-to-decide* if no PPT algorithm can distinguish random elements of $\mathcal{L}$ from random elements of $\mathcal{V}\backslash\mathcal{L}$. When speaking of a *hard language*, we mean a language which is hard-to-decide. For example, for fixed generators $(g, h) \in \mathbb{G}^2$ in a cyclic group $\mathbb{G}$, the Diffie-Hellman relation $R((g_1, g_2), w) := \big((g_1, g_2) = (g^w, h^w)\big)$ defines a hard-to-decide language in $\mathcal{V} = \mathbb{G}^2$ as long as the DDH assumption holds in $\mathbb{G}$.

Proving a statement $x \in \mathcal{L}$ can be done by demonstrating the existence of w such that that $R(x, w) = 1$. Also, the relation R can be defined so as to take as input a set of public parameters $\mathsf{pp} \leftarrow \mathsf{Setup}(\lambda)$, so that R is generated as $R \leftarrow \mathcal{R}(\mathsf{pp})$ rather than simply from the security parameter.

In non-interactive zero-knowledge proof systems, there is no online conversation between the prover and the verifier: each proof consists of a single message from the former to the latter. In addition to common public parameters pp, the prover and the verifier both take as input a common reference string crs which can be seen as another set of public parameters generated by a trusted party. In some cases, the public parameters pp can be part of the common reference string crs but it will be useful to separate them.

Definition 2 (NIZK Proofs [38, 37]). *A non-interactive zero-knowledge (NIZK) proof system $\Pi_{\mathcal{P}}$ for a family of hard relations $\mathcal{R}$ is a tuple of algorithms $(\mathsf{Setup}_{\mathcal{P}}, \mathsf{CRS\text{-}Gen}_{\mathcal{P}}, \mathsf{Prove}, \mathsf{Verify}_{\mathcal{P}})$.*

Setup$_{\mathcal{P}}(1^{\lambda})$*: from the security parameter λ, generates the public parameters* pp *of the proof system;*

CRS-Gen$_{\mathcal{P}}(\mathsf{pp})$*: takes in* pp *and outputs the common reference string* crs *that are public elements helping performing a proof for $R \leftarrow \mathcal{R}(\mathsf{pp})$*

Prove(crs, x, w)*: computes a proof π for x using the public* crs *and the private witness w.*

Verify$_{\mathcal{P}}(\mathsf{crs}, x, \pi)$*: returns either 1 or 0 if π is a valid proof associated to the language L_R.*

A NIZK proof system $\Pi_{\mathcal{P}}$ has the following properties:

Perfect Completeness: *for any PPT adversary $\mathcal{A}_1$,*

$$\Pr[\mathsf{pp} \leftarrow \mathsf{Setup}_{\mathcal{P}}(\lambda);\ \mathsf{crs} \leftarrow \mathsf{CRS\text{-}Gen}_{\mathcal{P}}(\mathsf{pp});\ (x, w) \leftarrow \mathcal{A}_1(\mathsf{crs});\\ \pi \leftarrow \mathsf{Prove}(\mathsf{crs}, x, w) : \mathsf{Verify}(\mathsf{crs}, x, \pi) = 0\ \wedge\ R(x, w) = 1] = 0,$$

Computational Soundness: *for any PPT adversary $\mathcal{A}_2$,*

$$\Pr[\mathsf{pp} \leftarrow \mathsf{Setup}_{\mathcal{P}}(\lambda);\ \mathsf{crs} \leftarrow \mathsf{CRS\text{-}Gen}_{\mathcal{P}}(\mathsf{pp});\ (x, \pi) \leftarrow \mathcal{A}_2(\mathsf{crs}) :\\ \mathsf{Verify}_{P}(\mathsf{crs}, x, \pi) = 1\ \wedge\ (\nexists w : R(x, w) = 1)] \in \mathsf{negl}(\lambda),$$

The notion of **statistical** *soundness is obtained by allowing $\mathcal{A}_2$ to be a computationally unbounded adversary. Non-interactive proof systems where the soundness property is only guaranteed in the computational sense are often called* **arguments**.

Zero-Knowledge: *there exists a PPT simulator $(\mathsf{S}_1, \mathsf{S}_2)$ such that, for any PPT adversary $\mathcal{A}_3$,*

$$\Pr[\mathsf{pp} \leftarrow \mathsf{Setup}_{\mathcal{P}}(\lambda);\ (\mathsf{crs}, \tau) \leftarrow \mathsf{S}_1(\mathsf{pp}) : \mathcal{A}_3^{\mathsf{S}_2(\mathsf{crs}, \tau, ., .)}(\mathsf{crs}) = 1]\\ \approx \quad \Pr[\mathsf{pp} \leftarrow \mathsf{Setup}_{\mathcal{P}}(\lambda);\ \mathsf{crs} \leftarrow \mathsf{CRS\text{-}Gen}_{\mathcal{P}}(\mathsf{pp}) : \mathcal{A}_3^{\mathcal{P}(\mathsf{crs}, ., .)}(\mathsf{crs}) = 1],$$

- $\mathcal{P}(\mathsf{crs}, ., .)$ *emulates the actual prover. It takes as input a pair (x, w) and outputs a proof π if $(x, w) \in R$. Otherwise, it outputs $\perp$,*

- $\mathsf{S}_2(\mathsf{crs}, \tau, ., .)$ *is an oracle that takes as input (x, w) and outputs a simulated proof $\pi \leftarrow \mathsf{S}_2(\mathsf{crs}, \tau, x)$ if $(x, w) \in R$ and $\perp$ if $(x, w) \notin R$. Importantly, π is computed without using the witness w if $(x, w) \in R$.*

In some cases, the public parameters are generated at the same time as the CRS crs by the **CRS-Gen**$_{\mathcal{P}}$ algorithm. The above definition allows them to be generated separately in order to capture Quasi-Adaptive NIZK proofs, which will be discussed later on.

The above definition of the zero-knowledge (ZK) property is computational since $\mathcal{A}_3$ is restricted to be efficient. By removing this restriction and allowing for an all powerful $\mathcal{A}_3$, we can capture statistical or perfect ZK if the distributions are statistically close or perfectly indistinguishable, respectively.

Intuitively, the zero-knowledge property captures that, for any $x \in \mathcal{L}$, the only information revealed by an honestly generated proof π is the same as a simulated proof that is generated without using w. Hence, the verifier learns nothing beyond the truth of the proven statement $x \in \mathcal{L}$. In particular, no information about the witness w is revealed. In many applications, a weaker notion called *witness-indistinguishability* suffices. It requires that, when a given statement $x \in \mathcal{L}$ admits at least two distinct witnesses w_0, w_1 such that $R(x, w_0) = R(x, w_1) = 1$, the distribution of a proof π for x does not depend on which witness is used to compute π. However, π may not be computable by an efficient simulator $(\mathsf{S}_1, \mathsf{S}_2)$ as in the zero-knowledge property.

Definition 3 (Witness Indistinguishability)**.** *A non-interactive proof system* $\Pi_{\mathcal{P}} = (\mathsf{Setup}_{\mathcal{P}}$, $\mathsf{CRS\text{-}Gen}_{\mathcal{P}}$, Prove, $\mathsf{Verify}_{\mathcal{P}})$ *for a hard language* $\mathcal{L}$ *is witness-indistinguishable (NIWI) if, for any PPT adversary* $(\mathcal{A}_4, \mathcal{A}_5)$*, for any* $\mathsf{pp} \leftarrow \mathsf{Setup}_{\mathcal{P}}(\lambda)$ *and* $\mathsf{crs} \leftarrow \mathsf{CRS\text{-}Gen}_{\mathcal{P}}(\mathsf{pp})$*,*

$$\begin{aligned} &\Pr[(x, w_0, w_1, \mathsf{st}) \leftarrow \mathcal{A}_4(\mathsf{crs}); \pi \leftarrow \mathsf{Prove}(\mathsf{crs}, x, w_0) : \mathcal{A}_5(\pi, \mathsf{st}) = 1] \\ &\qquad\qquad\qquad \approx \quad \Pr[(x, w_0, w_1, \mathsf{st}) \leftarrow \mathcal{A}_4(\mathsf{crs}); \pi \leftarrow \mathsf{Prove}(\mathsf{crs}, x, w_1) : \mathcal{A}_5(\pi, \mathsf{st}) = 1], \end{aligned}$$

where $(x, w_0), (x, w_1) \in R$.

For hard languages that admit efficient an zero-knowledge simulator, the latter can always use its simulation trapdoor to compute proof for true statements without knowing the witnesses. The trapdoor can also be used for computing proofs for false statements, *i. e.*proofs that satisfy the verification test although $x \notin \mathcal{L}$. This property is a very useful theoretic tool for building chosen-ciphertext-secure cryptosystems, for example based on the Naor-Yung/Sahai [208, 231] paradigm. The ability to simulate proofs for false statements should be used with caution as observing such fake proofs may help the adversary prove false statements by itself. The notion of simulation-soundness, as introduced by Sahai [231] captures that seeing a polynomial number of fake proofs should not break the soundness property.

Definition 4 (Simulation-Soundness [231])**.** *A non-interactive proof system* $\Pi_{\mathcal{P}} = (\mathsf{Setup}_{\mathcal{P}}$, $\mathsf{CRS\text{-}Gen}_{\mathcal{P}}$, Prove, $\mathsf{Verify}_{\mathcal{P}})$ *for a hard language* $\mathcal{L}$ *is simulation-sound if there exists a PPT simulator* $(\mathsf{S}_1, \mathsf{S}_2)$ *such that, for any PPT adversary* $\mathcal{A}_6$*,*

$$\begin{aligned} &\Pr[\mathsf{pp} \leftarrow \mathsf{Setup}_{\mathcal{P}}(\lambda);\ (\mathsf{crs}, \tau) \leftarrow \mathsf{S}_1(\mathsf{pp}); (x, \pi) \leftarrow \mathcal{A}_6^{\mathsf{S}_2(\mathsf{crs}, \tau, .,.)}(\mathsf{crs}) : \\ &\qquad\qquad \mathsf{Verify}_{P}(\mathsf{crs}, x, \pi) = 1 \ \wedge \ \neg(\exists w : R(x, w) = 1) \ \wedge \ (x, \pi) \notin Q] \in \mathsf{negl}(\lambda) \end{aligned}$$

where the adversary is granted access to an oracle $\mathsf{S}_2(\mathsf{crs}, \tau, .)$ *that takes as input a statement* x *(where* x *may be outside* $\mathcal{L}$*) and outputs a simulated proof* $\pi \leftarrow \mathsf{S}_2(\mathsf{crs}, \tau, x)$ *before setting* $Q := Q \cup \{(x, \pi)\}$*, which is initially empty.*

The proof system is said *unbounded* simulation-sound if it provides simulation-soundness against adversaries which are allowed to invoke the oracle $\mathsf{S}_2(\mathsf{crs}, \tau, .)$ an *a priori* unbounded (but polynomial) number of times. In the strictly weaker notion of *one-time* simulation-soundness, the adversary is restricted to query $\mathsf{S}_2(\mathsf{crs}, \tau, .)$ only once.

Note that, since proofs for false statements do exist in simulation-sound proof systems, the soundness property can only hold in the computational sense.

1.3 Groth-Sahai Proof Systems

In their seminal paper published in 2008, Groth and Sahai gave efficient non-interactive witness indistinguishable proof systems allowing to efficiently prove algebraic statements in groups with a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Their techniques build on earlier ideas suggested by Groth, Ostrovsky and Sahai [137, 136] in that they rely on homomorphic commitments that can be either perfectly hiding or perfectly binding depending on how the commitment key is generated. A difference with [137, 136], however, is that the Groth-Sahai methods directly demonstrate the validity of algebraic statements without proving the satisfiability of a circuit. While this restricts the range of provable languages, it allows for a much better efficiency as it avoids the need for an expensive NP reduction.

In Groth-Sahai proofs, the statements to be proved involve witnesses that can be either exponents in $\mathbb{Z}_p$ or group elements in $\mathbb{G}_1$ or $\mathbb{G}_2$. One caveat is that these NIWI proofs can only be used as proofs of knowledge when the witnesses are all group elements.

The Groth-Sahai (GS) proof systems can be instantiated using the K-linear assumption for any $K > 0$. In their instantiation based on the DLIN assumption (with $K = 2$) in symmetric pairing configurations (i.e., with $\mathbb{G}_1 = \mathbb{G}_2$), the Groth-Sahai (GS) proof systems [138] use a common reference string (CRS) consisting of three vectors $\mathbf{g_1}, \mathbf{g_2}, \mathbf{g_3} \in \mathbb{G}^3$, where $\mathbf{g_1} = (g_1, 1, g)$, $\mathbf{g_2} = (1, g_2, g)$ for some $g_1, g_2 \in \mathbb{G}$. In order to commit to a group element $X \in \mathbb{G}$, the prover computes $\mathbf{C} = (1, 1, X) \cdot \mathbf{g_1}^r \cdot \mathbf{g_2}^s \cdot \mathbf{g_3}^t$ with $r, s, t \xleftarrow{\$} \mathbb{Z}_p$. When the proof system is configured to provide perfectly sound proofs, $\mathbf{g_3}$ is set as $\mathbf{g_3} = \mathbf{g_1}^{\xi_1} \cdot \mathbf{g_2}^{\xi_2}$ with $\xi_1, \xi_2 \xleftarrow{\$} \mathbb{Z}_p$. In this case, commitments can be written as

$$\mathbf{C} = (g_1^{r+\xi_1 t}, g_2^{s+\xi_2 t}, X \cdot g^{r+s+t(\xi_1+\xi_2)}),$$

so that they can be interpreted as Boneh-Boyen-Shacham (BBS) ciphertexts. Moreover, the committed $X \in \mathbb{G}$ can be recovered by running the BBS decryption algorithm using the private key $(\alpha_1, \alpha_2) = (\log_g(g_1), \log_g(g_2))$. When the CRS is set up to give perfectly witness indistinguishable (WI) proofs, $\mathbf{g_1}, \mathbf{g_2}$ and $\mathbf{g_3}$ are linearly independent vectors, so that $\mathbf{C}$ is a perfectly hiding commitment to $X \in \mathbb{G}$: a typical choice is $\mathbf{g_3} = \mathbf{g_1}^{\xi_1} \cdot \mathbf{g_2}^{\xi_2} \cdot (1, 1, g)^{-1}$. Under the DLIN assumption, the two distributions of CRS are computationally indistinguishable.

To commit to an exponent $x \in \mathbb{Z}_p$, the prover computes $\mathbf{C} = \boldsymbol{\varphi}^x \cdot \mathbf{g_1}^r \cdot \mathbf{g_2}^s$, with $r, s \xleftarrow{\$} \mathbb{Z}_p$, using a CRS containing $\boldsymbol{\varphi}, \mathbf{g_1}, \mathbf{g_2}$. In the perfect soundness setting $\boldsymbol{\varphi}, \mathbf{g_1}, \mathbf{g_2}$ are linearly independent (typically $\boldsymbol{\varphi} = \mathbf{g_3} \cdot (1, 1, g)$ where $\mathbf{g_3} = \mathbf{g_1}^{\xi_1} \cdot \mathbf{g_2}^{\xi_2}$) whereas, in the perfect WI setting, choosing $\boldsymbol{\varphi} = \mathbf{g_1}^{\xi_1} \cdot \mathbf{g_2}^{\xi_2}$ yields perfectly hiding commitments since $\mathbf{C}$ is statistically independent of x.

To prove that committed variables satisfy a set of relations, the GS techniques replace variables by the corresponding commitments in each relation. The entire proof consists of one commitment per variable and one proof element (made of a constant number of elements) per relation.

Efficient NIWI proofs are available for pairing-product relations, which are equations of the type

$$\prod_{i=1}^{n} e(\mathcal{A}_i, \mathcal{X}_i) \cdot \prod_{i=1}^{n} \cdot \prod_{j=1}^{n} e(\mathcal{X}_i, \mathcal{X}_j)^{a_{ij}} = t_T, \tag{1.5}$$

for constants $t_T \in \mathbb{G}_T$, $\mathcal{A}_1, \ldots, \mathcal{A}_n \in \mathbb{G}$, $a_{ij} \in \mathbb{Z}_p$, for $i, j \in \{1, \ldots, n\}$, and variables $\mathcal{X}_1$, $\ldots, \mathcal{X}_n \in \mathbb{G}$. Efficient proofs also exist for multi-exponentiation equations, which are of the form

$$\prod_{i=1}^{m} \mathcal{A}_i^{y_i} \cdot \prod_{j=1}^{n} \mathcal{X}_j^{b_j} \cdot \prod_{i=1}^{m} \cdot \prod_{j=1}^{n} \mathcal{X}_j^{y_i \gamma_{ij}} = T,$$

for constants $T, \mathcal{A}_1, \ldots, \mathcal{A}_m \in \mathbb{G}, b_1, \ldots, b_n \in \mathbb{Z}_p$ and $\gamma_{ij} \in \mathbb{Z}_p$, for $i \in \{1, \ldots, m\}, j \in \{1, \ldots, n\}$ and variables $\mathcal{X}_1, \ldots, \mathcal{X}_n \in \mathbb{G}, y_1, \ldots, y_m \in \mathbb{Z}_p$.

Multi-exponentiation equations always admit non-interactive zero-knowledge (NIZK) proofs at no additional cost. On a perfectly witness indistinguishable CRS, a trapdoor (such as the hidden exponents $(\xi_1, \xi_2) \in \mathbb{Z}_p^2$ when $\mathbf{g_3} = \mathbf{g_1}^{\xi_1} \cdot \mathbf{g_2}^{\xi_2} \cdot (1, 1, g)^{-1}$) makes it possible to simulate proofs without knowing witnesses and simulated proofs are perfectly indistinguishable from real proofs. As for pairing-product equations, zero-knowledge proofs are often possible – this is usually the case when the right-hand-side member t_T of (1.5) is a product of pairings involving known group elements – but the number of group elements per proof may not be constant anymore. Here, when using such NIZK simulators, we just introduce a constant number of extra group elements in the proofs.

In both cases, proofs for quadratic equations cost 9 group elements. Linear pairing-product equations (when $a_{ij} = 0$ for all i, j) take 3 group elements each. Linear multi-exponentiation equations of the type $\prod_{j=1}^{n} \mathcal{X}_j^{b_j} = T$ (resp. $\prod_{i=1}^{m} \mathcal{A}_i^{y_i} = T$) demand 3 (resp. 2) group elements.

Groth-Sahai proofs can also be instantiated under the SXDH assumption. This instantiation uses prime order groups and a common reference string containing two vectors $\mathbf{f_1}, \mathbf{f_2} \in \mathbb{G}^2$, where $\mathbf{f_1} = (g, f_1)$, $\mathbf{f_2} = (h, f_2)$, for some $g, h, f_1, f_2 \in \mathbb{G}$. To commit to a group element $X \in \mathbb{G}$, the prover chooses $r, s \xleftarrow{\$} \mathbb{Z}_p$ and computes $\mathbf{C} = (1, X) \cdot \mathbf{f_1}^r \cdot \mathbf{f_2}^s$. On a perfectly sound common reference string, we have $\mathbf{f_2} = \mathbf{f_1}^{\xi}$, for some $\xi \in \mathbb{Z}_p$. Commitments $\mathbf{C} = (g^{r+\xi s}, f_1^{r+\xi s} \cdot X)$ are extractable as their distribution coincides with that of an Elgamal ciphertexts [103] and the committed X can be extracted using $\beta = \log_g(f_1)$. In the witness indistinguishability (WI) setting, the vector $\mathbf{f_2}$ is chosen so that $(\mathbf{f_1}, \mathbf{f_2})$ are linearly independent vectors and $\mathbf{C}$ is a perfectly hiding commitment. Under the DDH assumption in $\mathbb{G}$, the two kinds of CRS can be exchanged for one another without the adversary noticing.

To convince the verifier that committed variables satisfy a set of relations, the prover computes one commitment per variable and one proof element per equation.

In pairing-product equations, proving a linear equation of the form

$$\prod_{i=1}^{n} e(\mathcal{X}_i, \mathcal{A}_i) = t_T, \tag{1.6}$$

where $\mathcal{X}_1, \ldots, \mathcal{X}_n \in \mathbb{G}$ and $\mathcal{A}_1, \ldots, \mathcal{A}_n \in \hat{\mathbb{G}}$, costs two elements of $\hat{\mathbb{G}}$. If variables are in $\hat{\mathbb{G}}$, proofs must live in $\mathbb{G}^2$ instead of $\hat{\mathbb{G}}^2$. Quadratic equations are somewhat more expensive to prove and they main contain elements of both $\mathbb{G}$ and $\hat{\mathbb{G}}$. Multi-exponentiation equations have similar proof sizes.

In [28], Belenkiy *et al.* showed that Groth-Sahai proofs are perfectly randomizable. Given commitments $\{\mathbf{C}_{\mathcal{X}_i}\}_{i=1}^{n}$ and a NIWI proof $\boldsymbol{\pi}_{\text{PPE}}$ that committed $\{\mathcal{X}\}_{i=1}^{n}$ satisfy (1.5), anyone can publicly compute re-randomized commitments $\{\mathbf{C}_{\mathcal{X}_i'}\}_{i=1}^{n}$ and a re-randomized proof $\boldsymbol{\pi}'_{\text{PPE}}$ of the same statement. Moreover, $\{\mathbf{C}_{\mathcal{X}_i'}\}_{i=1}^{n}$ and π'_{PPE} are distributed as freshly generated commitments and proof.

Groth-Sahai proofs are also malleable [83] in that it is often possible to publicly modify a proof π of a given statement x and turn it into a proof π' of another statement x' which is related to x. This malleability property – which appears unique to GS proofs – can be a useful property in certain situations. For example, Belenkiy *et al.* used it to construct delegatable anonymous credentials [28]. More recently, Chase *et al.* [83] took advantage of the malleability of Groth-Sahai proofs to build homomorphic encryption schemes satisfying a relaxed form of chosen-ciphertext security [222], efficient non-interactive proofs for shuffles and elections systems [83, 84].

In the design of non-malleable protocols like chosen-ciphertext-secure public-key encryption, however, this malleability property is usually undesirable. Groth [133] showed an elegant technique, inspired by earlier ideas due to Lindell [192], for tweaking Groth-Sahai proofs and obtain unbounded simulation-soundness. The upcoming chapters will present more efficient methods for obtaining one-time and unbounded simulation-sound variants of Groth-Sahai proofs.

1.4 Quasi-Adaptive NIZK Proofs

While much more efficient than general NIZK proofs, the GS techniques remain more expensive than non-interactive proofs obtained from the Fiat-Shamir heuristic [107] in the random oracle model [32]: for example, proving that t variables satisfy a system of n linear equations demands $\Theta(t+n)$ group elements where Σ-protocols allow for $\Theta(t)$-size proofs.

For languages consisting of linear subspaces of a vector space, Jutla and Roy [151] showed how to significantly improve upon the GS paradigm in the *quasi-adaptive* setting. In quasi-adaptive NIZK proofs (QA-NIZK) for a class of languages $\{\mathcal{L}_\rho\}$ parametrized by ρ, the common reference string (CRS) is allowed to depend on the particular language $\mathcal{L}_\rho$ of which membership must be proved. At the same time, a single simulator should be effective for the whole class of languages $\{\mathcal{L}_\rho\}$. As pointed out in [151], QA-NIZK proofs are sufficient for many applications of Groth-Sahai proofs. In this setting, Jutla and Roy [151] gave very efficient QA-NIZK proofs of membership in linear subspaces. If $\mathbf{A} \in \mathbb{Z}_p^{t\times n}$ is a matrix or rank $t < n$, in order to prove membership of $\mathcal{L} = \{\mathbf{v} \in \mathbb{G}^n \mid \exists \mathbf{x} \in \mathbb{Z}_p^t \text{ s.t. } \mathbf{v} = g^{\mathbf{x}\cdot\mathbf{A}}\}$, the Jutla-Roy proofs only take $O(n-t)$ group elements – instead of $\Theta(n+t)$ in [138] – at the expense of settling for computational soundness.

Quasi-Adaptive NIZK (QA-NIZK) proofs are NIZK proofs where the CRS is allowed to depend on the specific language for which proofs have to be generated. The CRS is divided into a fixed part Γ, produced by an algorithm $\mathbb{K}_0$, and a language-dependent part ψ. However, there should be a single simulator for the entire class of languages.

Let λ be a security parameter. For public parameters Γ produced by $\mathbb{K}_0$, let $\mathcal{D}_\Gamma$ be a probability distribution over a collection of relations $\mathcal{R} = \{R_\rho\}$ parametrized by a string ρ with an associated language $\mathcal{L}_\rho = \{x \mid \exists w : R_\rho(x,w) = 1\}$.

We consider proof systems where the prover and the verifier both take a label lbl as additional input. For example, this label can be the message-carrying part of an Elgamal-like encryption. Formally, a tuple of algorithms $(\mathbb{K}_0, \mathbb{K}_1, \mathsf{P}, \mathsf{V})$ is a QA-NIZK proof system for $\mathcal{R}$ if there exists a PPT simulator $(\mathsf{S}_1, \mathsf{S}_2)$ such that, for any PPT adversaries $\mathcal{A}_1, \mathcal{A}_2$ and $\mathcal{A}_3$, we have the following properties:

Quasi-Adaptive Completeness:

$$\begin{aligned}&\Pr[\Gamma \leftarrow \mathbb{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ \psi \leftarrow \mathbb{K}_1(\Gamma,\rho);\\ &(x,w,\mathsf{lbl}) \leftarrow \mathcal{A}_1(\Gamma,\psi,\rho);\ \pi \leftarrow \mathsf{P}(\psi,x,w,\mathsf{lbl}) : \mathsf{V}(\psi,x,\pi,\mathsf{lbl}) = 1 \text{ if } R_\rho(x,w) = 1] = 1.\end{aligned}$$

Quasi-Adaptive Soundness:

$$\begin{aligned}&\Pr[\Gamma \leftarrow \mathbb{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ \psi \leftarrow \mathbb{K}_1(\Gamma,\rho);\ (x,\pi,\mathsf{lbl}) \leftarrow \mathcal{A}_2(\Gamma,\psi,\rho) :\\ &\qquad \mathsf{V}(\psi,x,\pi,\mathsf{lbl}) = 1 \ \wedge\ \neg(\exists w : R_\rho(x,w) = 1)] \in \mathsf{negl}(\lambda).\end{aligned}$$

Quasi-Adaptive Zero-Knowledge:

$$\begin{aligned}&\Pr[\Gamma \leftarrow \mathbb{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ \psi \leftarrow \mathbb{K}_1(\Gamma,\rho)\ :\ \mathcal{A}_3^{\mathsf{P}(\psi,.,.)}(\Gamma,\psi,\rho) = 1]\\ &\quad \approx \Pr[\Gamma \leftarrow \mathbb{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ (\psi,\tau_{sim}) \leftarrow \mathsf{S}_1(\Gamma,\rho)\ :\ \mathcal{A}_3^{\mathsf{S}(\psi,\tau_{sim},.,.,.)}(\Gamma,\psi,\rho) = 1],\end{aligned}$$

where

- $\mathsf{P}(\psi,.,.,.)$ emulates the actual prover. It takes as input (x,w) and lbl and outputs a proof π if $(x,w) \in R_\rho$. Otherwise, it outputs $\perp$.
- $\mathsf{S}(\psi,\tau_{sim},.,.,.)$ is an oracle that takes as input (x,w) and lbl. It outputs a simulated proof $\mathsf{S}_2(\psi,\tau_{sim},x,\mathsf{lbl})$ if $(x,w) \in R_\rho$ and $\perp$ if $(x,w) \notin R_\rho$.

We assume that the CRS ψ contains an encoding of ρ, which is thus available to V. The definition of Quasi-Adaptive Zero-Knowledge requires a single simulator for the entire family of relations $\mathcal{R}$.

It is often useful to have a property called *simulation-soundness*, which requires that the adversary be unable to prove false statements even after having seen simulated proofs for possibly false statements.

Unbounded Simulation-Soundness: For any PPT adversary $\mathcal{A}_4$, it holds that

$$\begin{aligned}&\Pr[\Gamma \leftarrow \mathbb{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ (\psi,\tau_{sim}) \leftarrow \mathsf{S}_1(\Gamma,\rho);\ (x,\pi,\mathsf{lbl}) \leftarrow \mathcal{A}_4^{\mathsf{S}_2(\psi,\tau_{sim},.,.)}(\Gamma,\psi,\rho)\ :\\ &\qquad \mathsf{V}(\psi,x,\pi,\mathsf{lbl}) = 1 \ \wedge\ \neg(\exists w : R_\rho(x,w) = 1)\ \wedge\ (x,\pi,\mathsf{lbl}) \notin Q] \in \mathsf{negl}(\lambda),\end{aligned}$$

where the adversary is allowed unbounded access to an oracle $\mathsf{S}_2(\psi,\tau,.,.)$ that takes as input statement-label pairs (x,lbl) (where x may be outside $\mathcal{L}_\rho$) and outputs simulated proofs $\pi \leftarrow \mathsf{S}_2(\psi,\tau_{sim},x,\mathsf{lbl})$ before updating the set $Q = Q \cup \{(x,\pi,\mathsf{lbl})\}$, which is initially empty.

In the weaker notion of one-time simulation-soundness, only one query to the S_2 oracle is allowed.

In some applications, one may settle for a weaker notion, called *relative soundness* by Jutla and Roy [150], which allows for more efficient proofs, especially in the single-theorem case. Informally, relatively sound proof systems involve both a public verifier *and* a private verification algorithm, which has access to a trapdoor. For hard languages, the two verifiers should almost always agree on any adversarially-created proof. Moreover, the private verifier should not accept a non-trivial proof for a false statement, even if the adversary has already seen proofs for false statements.

A labeled single-theorem relatively sound QA-NIZK proof system is comprised of a quasi-adaptive labeled proof system $(\mathbb{K}_0, \mathbb{K}_1, \mathsf{P}, \mathsf{V})$ along with an efficient private verifier W and an efficient simulator $(\mathsf{S}_1, \mathsf{S}_2)$. Moreover, the following properties should hold for any PPT adversaries $(\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4)$.

Quasi Adaptive Relative Single-Theorem Zero-Knowledge:

$$\begin{aligned}
&\Pr[\Gamma \leftarrow \mathbb{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ \psi \leftarrow \mathbb{K}_1(\Gamma,\rho);\ (x,w,\mathsf{lbl},s) \leftarrow \mathcal{A}_1^{\mathsf{V}(\psi,\cdot,\cdot)}(\Gamma,\psi,\rho);\\
&\qquad\qquad \pi \leftarrow \mathsf{P}(\psi,\rho,x,w,\mathsf{lbl}) : \mathcal{A}_2^{\mathsf{V}(\psi,\cdot,\cdot)}(\pi,s) = 1]\\
&\approx \Pr[\Gamma \leftarrow \mathbb{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ (\psi,\tau) \leftarrow \mathsf{S}_1(\Gamma,\rho);\ (x,w,\mathsf{lbl},s) \leftarrow \mathcal{A}_1^{\mathsf{W}(\psi,\tau,\cdot,\cdot)}(\Gamma,\psi,\rho);\\
&\qquad\qquad \pi \leftarrow \mathsf{S}_2(\psi,\rho,\tau,x,\mathsf{lbl}) : \mathcal{A}_2^{\mathsf{W}(\psi,\tau,\cdot,\cdot)}(\pi,s) = 1],
\end{aligned}$$

Here, $\mathcal{A}_1$ is restricted to choosing (x,w) such that $R_\rho(x,w) = 1$.

Quasi Adaptive Relative Single-Theorem Simulation-Soundness:

$$\begin{aligned}
&\Pr[\Gamma \leftarrow \mathbb{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ (\psi,\tau) \leftarrow \mathsf{S}_1(\Gamma,\rho);\ (x,\mathsf{lbl},s) \leftarrow \mathcal{A}_3^{\mathsf{W}(\psi,\tau,\cdot,\cdot)}(\Gamma,\psi,\rho);\\
&\qquad \pi \leftarrow \mathsf{S}_2(\psi,\rho,\tau,x,\mathsf{lbl}) : (x',\mathsf{lbl}',\pi') \leftarrow \mathcal{A}_4^{\mathsf{W}(\psi,\tau,\cdot,\cdot)}(s,\pi) :\\
&(x,\pi,\mathsf{lbl}) \neq (x',\pi',\mathsf{lbl}') \wedge\ \nexists w' \text{ s.t. } R_\rho(x',w') = 1 \wedge \mathsf{W}(\psi,\tau,x',\mathsf{lbl}',\pi') = 1] \in \mathsf{negl}(\lambda)
\end{aligned}$$

Note that the definition of relative simulation-soundness does not require the adversary to provide a witness but the definition of single-theorem zero-knowledge does.

1.5 Structure-Preserving Cryptography

Many anonymity-related cryptographic protocols (e.g., [81, 6, 4, 112, 5, 2]) build on Groth-Sahai proofs in order to prove security in the standard model of computation. In order to guarantee the extractability of witnesses for proofs generated on a perfectly sound CRS, it is convenient to have signature schemes which allow one to sign elements of bilinear groups while maintaining the feasibility of conveniently proving that a committed signature is valid for a committed message.

Signature schemes where messages only consist of group elements appeared for the first time as ingredients of Groth's construction [133] of group signatures in the standard model. The scheme of [133] was mostly a proof of concept, with signatures consisting of thousands of group elements. More efficient solutions were described by Fuchsbauer [112] and, independently, in a paper of mine [81]. While the scheme of [112] is somewhat more efficient, it only allows signing messages with a particular structure (typically, Diffie-Hellman tuples). The construction of Cathalo, Yung and myself [81] does not have this restriction but its disadvantage resides in the linear length $O(n)$ of signatures if $\mathbb{G}^n$ is the message space. Abe, Haralambiev and Ohkubo [6, 4] – who introduced the "structure-preserving" terminology – subsequently showed how to sign messages of n group elements at once using $O(1)$-size signatures. Lower bounds on the size of structure-preserving signatures were given in [5] while

Abe *et al.* [5] provided evidence that optimally short SPS necessarily rely on interactive assumptions. As an ingredient for their tightly secure cryptosystems, Hofheinz and Jager [142] gave constructions based on the Decision Linear assumption [44] while similar results were independently achieved in [63, 82]. Quite recently, Abe *et al.* [2, 3] obtained constant-size signatures without sacrificing the security guarantees offered by security proofs under simple assumptions.

In the context of symmetric pairings, the description below assumes public parameters $\mathsf{pp} = \big((\mathbb{G}, \mathbb{G}_T),\, g\big)$ consisting of bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$, where $\lambda \in \mathbb{N}$ and a generator $g \in \mathbb{G}$.

Keygen(pp, n)**:** given an upper bound $n \in \mathbb{N}$ on the number of group elements per signed message, choose generators $G_r, H_r \stackrel{\$}{\leftarrow} \mathbb{G}$. Pick $\gamma_z, \delta_z \stackrel{\$}{\leftarrow} \mathbb{Z}_p$ and $\gamma_i, \delta_i \stackrel{\$}{\leftarrow} \mathbb{Z}_p$, for $i = 1$ to n. Then, compute $G_z = G_r^{\gamma_z}$, $H_z = H_r^{\delta_z}$ and $G_i = G_r^{\gamma_i}$, $H_i = H_r^{\delta_i}$ for each $i \in \{1, \ldots, n\}$. Finally, choose $\alpha_a, \alpha_b \stackrel{\$}{\leftarrow} \mathbb{Z}_p$ and define $A = e(G_r, g^{\alpha_a})$ and $B = e(H_r, g^{\alpha_b})$. The public key is defined to be

$$pk = \big(G_r,\ H_r,\ G_z,\ H_z,\ \{G_i, H_i\}_{i=1}^n,\ A,\ B\big) \in \mathbb{G}^{2n+4} \times \mathbb{G}_T^2$$

while the private key is $sk = \big(\alpha_a, \alpha_b, \gamma_z, \delta_z, \{\gamma_i, \delta_i\}_{i=1}^n\big)$.

Sign$(sk, (M_1, \ldots, M_n))$**:** to sign $(M_1, \ldots, M_n) \in \mathbb{G}^n$ using $sk = (\alpha_a, \alpha_b, \gamma_z, \delta_z, \{\gamma_i, \delta_i\}_{i=1}^n)$, choose $\zeta, \rho_a, \rho_b, \omega_a, \omega_b \stackrel{\$}{\leftarrow} \mathbb{Z}_p$ and compute $\theta_1 = g^\zeta$ as well as

$$\theta_2 = g^{\rho_a - \gamma_z \zeta} \cdot \prod_{i=1}^n M_i^{-\gamma_i}, \qquad \theta_3 = G_r^{\omega_a}, \qquad \theta_4 = g^{(\alpha_a - \rho_a)/\omega_a},$$

$$\theta_5 = g^{\rho_b - \delta_z \zeta} \cdot \prod_{i=1}^n M_i^{-\delta_i}, \qquad \theta_6 = H_r^{\omega_b}, \qquad \theta_7 = g^{(\alpha_b - \rho_b)/\omega_b},$$

The signature consists of $\sigma = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7)$.

Verify$(pk, \sigma, (M_1, \ldots, M_n))$**:** parse σ as $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7) \in \mathbb{G}^7$ and return 1 iff these equalities hold:

$$\begin{aligned} A &= e(G_z, \theta_1) \cdot e(G_r, \theta_2) \cdot e(\theta_3, \theta_4) \cdot \prod_{i=1}^n e(G_i, M_i), \\ B &= e(H_z, \theta_1) \cdot e(H_r, \theta_5) \cdot e(\theta_6, \theta_7) \cdot \prod_{i=1}^n e(H_i, M_i). \end{aligned}$$

The scheme was proved [6, 4] existentially unforgeable under chosen-message attacks under the q-SFP assumption, where q is the number of signing queries.

As shown in [6, 4], Signatures components $\{\theta_i\}_{i=2}^7$ can be publicly randomized to obtain a different signature $\{\theta_i'\}_{i=1}^7 \leftarrow \mathsf{ReRand}(pk, \sigma)$ on $(M_1, \ldots, M_n)$. After randomization, we have $\theta_1' = \theta_1$ whereas other signature components $\{\theta_i'\}_{i=2}^7$ are uniformly distributed among the values satisfying the relations

$$\begin{aligned} e(G_r, \theta_2') \cdot e(\theta_3', \theta_4') &= e(G_r, \theta_2) \cdot e(\theta_3, \theta_4) \\ e(H_r, \theta_5') \cdot e(\theta_6', \theta_7') &= e(H_r, \theta_5) \cdot e(\theta_6, \theta_7). \end{aligned}$$

Moreover, $\{\theta_i'\}_{i\in\{3,4,6,7\}}$ are statistically independent of the message and the rest of the signature. This implies that, in privacy-preserving protocols, re-randomized $\{\theta_i'\}_{i\in\{3,4,6,7\}}$ can safely appear in clear as long as $(M_1,\ldots,M_n)$ and $\{\theta_i'\}_{i\in\{1,2,5\}}$ are given in committed form.

In [5], Abe, Groth, Haralambiev and Ohkubo described shorter structure-preserving signatures based on interactive assumptions (or, alternatively, in the generic group model [237]). In the forthcoming chapters, we only rest on non-interactive and falsifiable assumptions, so that the above scheme will be preferred to those of [5].

In [2, 3], Abe *et al.* described constant-size structure-preserving signatures based on the standard DLIN assumption. While these constructions allow for the modular design of many privacy-enhancing protocols (e.g., group signatures) based on simple assumptions, they are somewhat less efficient than the original AHO signature [6]. While several of our results build on the latter system (as they were published before [2, 3]), they can often be modified by using the DLIN-based structure-preserving signatures of [2, 3] so as to avoid non-standard q-type assumption.

Regarding primitives beyond signature schemes, Camenisch *et al.* [65] showed a structure-preserving variant of the Cramer-Shoup cryptosystem [88] and used it to implement oblivious third parties [64]. Groth [135] described length-reducing trapdoor commitments (*i. e.*, where the commitment is shorter than the committed message) to group elements whereas [7] showed the impossibility of realizing such commitments when the commitment string lives in the same group as the message. Sakai *et al.* [233] recently suggested to use structure-preserving identity-based encryption [236] systems to restrict the power of the opening authority in group signatures.

CHAPTER 2

Applications of Structure-Preserving Cryptography and NIZK Proofs to Privacy-Enhancing Primitives

This chapter presents two applications of structure-preserving cryptography and Groth-Sahai proofs in the setting of privacy-preserving protocols where users can retain anonymity while taking certain actions within a group they belong to.

The first application is the design of a non-interactive group encryption system [156], where anyone can encrypt a message for a certified but anonymous member of a group of users. At the same time, the sender can convince anyone that a ciphertext is a valid encryption intended for some group member which an authority can identify if necessary.

The second application deals with the revocation problem in group signatures. Group signatures [85] are signatures schemes where group users can sign messages while hiding their identify within a group of members. Again, in order to deter abuses of the system, an authority is capable of identifying the author of any signature.

In this chapter, although group signatures are an older primitive than group encryption, our result on group encryption will be presented first since it makes use of our realization of structure-preserving signatures [81], which is less efficient than the one of Abe *et al.* [6] that we use in our revocable group signatures [180, 179].

2.1 Non-Interactive Group Encryption

Introduced by Kiayias, Tsiounis and Yung [156], group encryption (GE) is the encryption analogue of group signatures [85]. The latter primitives allow a group member to sign messages in the name of a group without revealing his identity. In a similar spirit, GE systems aim to hide the identity of a ciphertext's recipient and still guarantee that he belongs to a population of registered members in a group administered by a group manager (GM). A sender can generate an anonymous encryption of some plaintext m intended for a receiver holding a public key that was certified by the GM (message security and receiver anonymity being both in the CCA2 sense). The ciphertext is prepared while leaving an opening authority (OA) the ability to "open" the ciphertext (analogously to the opening operation in group signatures) and uncover the receiver's name. At the same time, the sender should be able to convince a verifier that: (1) The ciphertext is a valid encryption under the public key of some group member holding a valid certificate; (2) If necessary, the opening authority will be able to find out who the receiver is; (3) The plaintext is a witness satisfying some public relation.

MOTIVATIONS. As a natural use case, group encryption allows a firewall to block all encrypted emails attempting to enter a network unless they are generated for some certified organization member and they carry a proof of malware-freeness. Group encryption also enables oblivious retriever storage mechanisms in the cloud. Namely, when encrypting datasets on a remote storage server, the sender can convince this server that the data is intended for some legitimate certified user (who paid a subscription for storing his data) without disclosing the latter's identity. The GE primitive was also motivated by various privacy applications such as anonymous trusted third parties. Many cryptographic protocols such as fair exchange, fair encryption or escrow encryption, involve trusted third parties that remain offline most of the time and are only involved to resolve problems. Group encryption allows one to verifiably encrypt some message to such a trusted third party while hiding his identity among a set of possible trustees. For instance, a user can encrypt a key (e.g., in an "international key escrow system") to his own national trusted representative without letting the ciphertext reveal the latter's identity, which could leak information on the user's citizenship. At the same time, everyone can be convinced that the ciphertext is heading for an authorized trustee.

Group encryption also finds applications in ubiquitous computing, where anonymous credentials must be transferred between peer devices belonging to the same group. Asynchronous transfers may require to involve an untrusted storage server to temporarily store encrypted credentials. In such a situation, GE schemes may be used to simultaneously guarantee that (1) the server retains properly encrypted valid credentials that it cannot read; (2) credentials have a legitimate anonymous retriever; (3) if necessary, an authority will be able to determine who the retriever is.

By combining cascaded group encryptions using multiple trustees and according to a sequence of identity discoveries and transfers, one can also implement group signatures where signers can flexibly specify how a set of trustees should operate to open their signatures.

PRIOR WORKS. Kiayias, Tsiounis and Yung (KTY) [156] formalized the concept of group encryption and gave a suitable security modeling. They presented a modular design of GE system and proved that, beyond zero-knowledge proofs, anonymous public key encryption schemes with CCA2 security, digital signatures, and equivocal commitments are necessary to realize the primitive. They also showed how to efficiently instantiate their general construction using Paillier's cryptosystem [216]. While efficient, their scheme is not a single message encryption, since it requires the sender to interact with the verifier in a Σ-protocol to convince him that the aforementioned properties are satisfied. Interaction can be removed using the Fiat-Shamir paradigm [107] (and thus the random oracle model [32]), but only heuristic arguments [128] (see also [72]) are then possible in terms of security.

Independently, Qin *et al.* [224] considered a closely related primitive with non-interactive proofs and short ciphertexts. However, they avoid interaction by employing a random oracle and also rely on strong interactive assumptions. As we can see, none of these schemes is a truly non-interactive encryption scheme without the random oracle idealization.

OUR CONTRIBUTION. As already noted in various contexts such as anonymous credentials [29], rounds of interaction are expensive and even impossible at times as, in some applications, proofs should be verifiable by third parties that are not present when provers are available. In the setting of group encryption, this last concern is even more constraining as it requires the sender, who may be required to repeat proofs with many verifiers, to maintain a state and remember the random coins that he uses to encrypt *every* single ciphertext. In the frequent situation where many encryptions have to be generated using independent random coins, this becomes a definite bottleneck.

Together with Julien Cathalo and Moti Yung [81], we solved the above problems and described the first realization of fully non-interactive group encryption with CCA2-security and anonymity in the standard model. In our scheme, senders do not need to maintain a state: thanks to the Groth-Sahai [138] non-interactive proof systems, the proof of a ciphertext can be generated once-and-for-all at the same time as the ciphertext itself. Furthermore, using suitable parameters and for a comparable security level, we can also shorten ciphertexts by a factor of 2 in comparison with the KTY scheme. As far as communication goes, the size of proofs allows decreasing by more than 75% the number of transmitted bits between the sender and the verifier.

Since our goal is to avoid interaction, we also design a joining protocol (*i.e.*, a protocol whereby the user effectively becomes a group member and gets his public key certified by the GM) which requires the smallest amount of interaction: as in the Kiayias-Yung group signature [157], only two messages have to be exchanged between the GM and the user and the latter need not to prove anything about his public key. In particular, rewinding is not necessary in security proofs and the join protocol can be safely executed in a concurrent environment, when many users want to register at the same time. The join protocol uses a non-interactive public key certification scheme where discrete-logarithm-type public keys can be signed as if they were ordinary messages (and without knowing the matching private key) while leaving the ability to efficiently prove knowledge of the certificate/public key using the Groth-Sahai techniques. To certify users without having to rewind[1] in security proofs, the KTY scheme uses groups of hidden order (and more precisely, Camenisch-Lysyanskaya signatures [68]). In public order groups, to the best of our knowledge, our construction is the first certification method that does not require any form of proof of knowledge of private keys. We believe it to be of independent interest as it can be used to construct group signatures (in the standard model) where the joining mechanism tolerates concurrency in the model of [157] without demanding more than two moves of interaction.

2.1.1 Model and Security Notions

Syntax. Group encryption schemes involve a sender, a verifier, a group manager (GM) that manages the group of receivers and an opening authority (OA) which is able to uncover the identity of ciphertext receivers. A GE system is formally specified by the description of a relation $\mathcal{R}$ as well as a collection $\mathsf{GE} = \big(\mathsf{SETUP}, \mathsf{JOIN}, \langle \mathcal{G}_r, \mathcal{R}, \mathsf{sample}_{\mathcal{R}} \rangle, \mathsf{ENC}, \mathsf{DEC}, \mathsf{OPEN}, \langle \mathcal{P}, \mathcal{V} \rangle\big)$ of algorithms or protocols. Among these, SETUP is a set of initialization procedures that all take (explicitly or implicitly) a security parameter λ as input. They can be split into one that generates a set of public parameters params (a common reference string), one for the GM and another one for the OA. We call them $\mathsf{SETUP}_{\mathsf{init}}(\lambda)$, $\mathsf{SETUP}_{\mathsf{GM}}(\mathsf{params})$ and $\mathsf{SETUP}_{\mathsf{OA}}(\mathsf{params})$, respectively. The latter two procedures are used to produce key pairs $(\mathsf{pk}_{\mathsf{GM}}, \mathsf{sk}_{\mathsf{GM}})$, $(\mathsf{pk}_{\mathsf{OA}}, \mathsf{sk}_{\mathsf{OA}})$ for the GM and the OA. In the following, params is incorporated in the inputs of all algorithms although we sometimes omit to explicitly write it.

$\mathsf{JOIN} = (\mathsf{J}_{\mathsf{user}}, \mathsf{J}_{\mathsf{GM}})$ is an interactive protocol between the GM and the prospective user.

[1] Although the simulator does not need to rewind proofs of knowledge in [156], users still have to interactively prove the validity of their public key.

As in [157], we will restrict this protocol to have minimal interaction and consist of only two messages: the first one is the user's public key pk sent by $\mathsf{J}_{\mathsf{user}}$ to J_{GM} and the latter's response is a certificate $\mathsf{cert}_{\mathsf{pk}}$ for pk that makes the user's group membership effective. We do not require the user to prove knowledge of his private key sk or anything else about it. In our construction, valid keys will be publicly recognizable and users do not need to prove their validity. After the execution of JOIN, the GM stores the public key pk and its certificate $\mathsf{cert}_{\mathsf{pk}}$ in a public directory $\mathsf{database}$.

Algorithm sample allows sampling pairs $(x, w) \in \mathcal{R}$ (made of a public value x and a witness w) using keys $(\mathsf{pk}_{\mathcal{R}}, \mathsf{sk}_{\mathcal{R}})$ produced by $\mathcal{G}_r$. Depending on the relation, $\mathsf{sk}_{\mathcal{R}}$ may be the empty string (as will be the case in our scheme). The testing procedure $\mathcal{R}(x, w)$ returns 1 whenever $(x, w) \in \mathcal{R}$. To encrypt a witness w such that $(x, w) \in \mathcal{R}$ for some public x, the sender fetches the pair $(\mathsf{pk}, \mathsf{cert}_{\mathsf{pk}})$ from $\mathsf{database}$ and runs the randomized encryption algorithm. The latter takes as input w, a label L, the receiver's pair $(\mathsf{pk}, \mathsf{cert}_{\mathsf{pk}})$ as well as public keys $\mathsf{pk}_{\mathsf{GM}}$ and $\mathsf{pk}_{\mathsf{OA}}$. Its output is a ciphertext $\psi \leftarrow \mathsf{ENC}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}_{\mathsf{pk}}, w, L)$. On input of the same elements, the certificate $\mathsf{cert}_{\mathsf{pk}}$, the ciphertext ψ and the random coins $coins_{\psi}$ that were used to produce it, the non-interactive algorithm P generates a proof π_{ψ} that there exists a certified receiver whose public key was registered in $\mathsf{database}$ and that is able to decrypt ψ and obtain a witness w such that $(x, w) \in \mathcal{R}$. The verification algorithm V takes as input ψ, $\mathsf{pk}_{\mathsf{GM}}$, $\mathsf{pk}_{\mathsf{OA}}$, π_{ψ} and the description of $\mathcal{R}$ and outputs 0 or 1. Given ψ, L and the receiver's private key sk, the output of DEC is either a witness w such that $(x, w) \in \mathcal{R}$ or a rejection symbol $\perp$. Finally, OPEN takes as input a ciphertext/label pair (ψ, L) and the OA's secret key $\mathsf{sk}_{\mathsf{OA}}$ and returns a receiver's public key pk.

Security notions. The security model of Kiayias, Tsiounis and Yung [156] considers three notions called message security, anonymity and soundness. The first one captures the CCA2-security of messages encrypted under the receiver's public key, even if the adversary controls both the group manager and the opening authority. The notion of anonymity subsumes the anonymity of group encryption ciphertexts (in particular, the inability to tell apart encryptions of ciphertexts encrypted under pk_0 from those encrypted under pk_1), even given access to an opening oracle (run on behalf of the opening authority) and decryption oracles for both pk_0 and pk_1. The notion of soundness captures the security of the group manager against malicious encryptors colluding with a dishonest opening authority. In short, no malicious sender (even with the help of a corrupted opening authority) can create a valid proof for a ciphertext whose receiver cannot be traced to a certified group member. Detailed definitions are given in [156, 81]

2.1.2 Building Blocks: Structure-Preserving Commitments and Signatures

Our structure-preserving signature uses a trapdoor commitment to group elements as an important ingredient to dispense with proofs of knowledge of users' private keys.

A Strictly Structure-Preserving Trapdoor Commitment

We need a trapdoor commitment scheme that allows committing to elements of a group $\mathbb{G}$ where bilinear map arguments are taken. The scheme has to be structure-preserving in the strict sense in that commitments will have to be themselves elements of $\mathbb{G}$, which prevents us from using Groth's scheme [135] where commitments live in the range $\mathbb{G}_T$ of the pairing.

Such commitments can be obtained using the perfectly hiding Groth-Sahai commitment based on the linear assumption recalled in section 1.3. This commitment scheme uses a common reference string describing a prime order group $\mathbb{G}$ and a generator $f \in \mathbb{G}$. The commitment key consists of vectors $(\mathbf{f_1}, \mathbf{f_2}, \mathbf{f_3})$ chosen as $\mathbf{f_1} = (f_1, 1, f)$, $\mathbf{f_2} = (1, f_2, f)$ and $\mathbf{f_3} = \mathbf{f_1}^{\xi_1} \cdot \mathbf{f_2}^{\xi_2} \cdot (1, 1, f)^{\xi_3}$, with $f_1, f_2 \stackrel{\$}{\leftarrow} \mathbb{G}$, $\xi_1, \xi_2, \xi_3 \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$. To commit to a group element $X \in \mathbb{G}$, the sender picks $\phi_1, \phi_2, \phi_3 \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$ and sets $\mathbf{C}_X = (1, 1, X) \cdot \mathbf{f_1}^{\phi_1} \cdot \mathbf{f_2}^{\phi_2} \cdot \mathbf{f_3}^{\phi_3}$, which, if $\mathbf{f_3}$ is parsed as $(f_{3,1}, f_{3,2}, f_{3,3})$, can be written $\mathbf{C}_X = (f_1^{\phi_1} \cdot f_{3,1}^{\phi_3}, f_2^{\phi_2} \cdot f_{3,2}^{\phi_3}, X \cdot f^{\phi_1+\phi_2} \cdot f_{3,3}^{\phi_3})$. Due to the use of GS proofs, commitment openings need to only consist of group elements (and no scalar). To open $\mathbf{C}_X = (C_1, C_2, C_3)$, the sender reveals $(D_1, D_2, D_3) = (f^{\phi_1}, f^{\phi_2}, f^{\phi_3})$ and X. The receiver is convinced that the committed value was X by checking that

$$\begin{cases} e(C_1, f) = e(f_1, D_1) \cdot e(f_{3,1}, D_3) \\ e(C_2, f) = e(f_2, D_2) \cdot e(f_{3,2}, D_3) \\ e(C_3, f) = e(X \cdot D_1 \cdot D_2, f) \cdot e(f_{3,3}, D_3). \end{cases}$$

If a cheating committer can produce distinct openings of $\mathbf{C}_X$, we can solve a SDP instance $(g_1, g_2, g_{1,c}, g_{2,d})$. Namely, the commitment key is set as $(f_1, f_2, f_{3,1}, f_{3,2}) = (g_1, g_2, g_{1,c}, g_{2,d})$ and $f, f_{3,3}$ are chosen at random. When the adversary outputs openings $(X, (D_1, D_2, D_3))$ and $(X', (D_1', D_2', D_3'))$, these openings must simultaneously satisfy the equalities

$$e(f_1, D_1/D_1') = e(f_{3,1}, D_3'/D_3), \qquad e(f_2, D_2/D_2') = e(f_{3,2}, D_3'/D_3)$$

and $e((XD_1D_2)/(X'D_1'D_2'), f) = e(f_{3,3}, D_3'/D_3)$. A solution to the SDP instance is obtained as $(u, v, w) = (D_1/D_1', D_2/D_2', D_3'/D_3)$, which is a non-trivial triple as long as $X' \neq X$.

We also observe that, using the trapdoor (ξ_1, ξ_2, ξ_3), the receiver can equivocate commitments. Given a commitment $\mathbf{C}_X$ and its opening $(X, (D_1, D_2, D_3))$, one can trapdoor open $\mathbf{C}_X$ to any other $X' \in \mathbb{G}$ (and without knowing $\log_g(X')$) by computing

$$D_1' = D_1 \cdot (X'/X)^{\xi_1/\xi_3}, \qquad D_2' = D_2 \cdot (X'/X)^{\xi_2/\xi_3}, \qquad D_3' = (X/X')^{1/\xi_3} \cdot D_3.$$

Unlike Groth's trapdoor commitment to group elements [135], the above construction is not length-reducing in that the commitment string is longer than the message. In *strictly* structure-preserving commitments (i.e., where the commitment lives in the source group $\mathbb{G}$ instead of the target group $\mathbb{G}_T$), however, Abe, Haralambiev and Ohkubo showed [7] that this is inevitable. A slightly more efficient construction of strictly structure-preserving trapdoor commitment was given in [7].

A Structure-Preserving Signature Scheme

In [81], we first described a structure-preserving signature scheme in order to certify public keys for the DLIN-based variant [235, 143] of the Cramer-Shoup cryptosystem [88, 90]. These keys should be signed while retaining algebraic properties that make it possible to prove knowledge of a public key and its corresponding certificate in an efficient way. In particular, signing hashed public keys is proscribed as it would destroy their algebraic structure. In the interactive setting, several papers (e.g., [39, 134]) described efficient interactive protocols where a public key is jointly generated by a user and a certification authority in such a way that the user eventually obtains a certified public key and no one else learns the underlying private key. In our construction, we aim at minimizing the amount of interaction and let users generate their public key entirely on their own before requesting their certification. Ideally, we would like to be able to sign public keys without even requiring users to prove knowledge of their private key and, in particular, without having to first rewind a proof of knowledge so as to extract the user's private key in the security proof. This is where structure-preserving signatures come in handy.

In the description, we assume common public parameters cp consisting of bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$, for a security parameter λ, and a generator $g \stackrel{\$}{\leftarrow} \mathbb{G}$. We also assume that certified public keys always consist of a fixed number n of group elements (*i.e.*, $\mathcal{PK} = \mathbb{G}^n$).

The scheme borrows from the Boyen-Waters group signature [56] in the use of the Hidden Strong Diffie-Hellman assumption. A simplified version of this scheme involves a signer that holds a public key $PK = (\Omega = g^\omega, A = (g,g)^\alpha, u, u_0, u_1 = g^{\beta_1}, \ldots, u_n = g^{\beta_n})$, for private elements $SK = (\omega, \alpha, \beta_1, \ldots, \beta_n)$, where n denotes the number of groups elements that certified public keys consist of. To certify a public key $\mathsf{pk} = (X_1 = g^{x_1}, \ldots, X_n = g^{x_n})$, the signer chooses an exponent $c_{\mathcal{ID}} \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$ and computes $S_1 = (g^\alpha)^{1/(\omega + c_{\mathcal{ID}})}$, $S_2 = g^{c_{\mathcal{ID}}}$, $S_3 = u^{c_{\mathcal{ID}}}$, $S_4 = (u_0 \cdot \prod_{i=1}^n X_i^{\beta_i})^{c_{\mathcal{ID}}}$ and $S_5 = (S_{5,1}, \ldots, S_{5,n}) = (X_1^{c_{\mathcal{ID}}}, \ldots, X_n^{c_{\mathcal{ID}}})$. Verification then checks whether $e(S_1, \Omega \cdot S_2) = A$ and $e(S_2, u) = e(g, S_3)$ as in [56]. It must also be checked that $e(S_4, g) = e(u_0, S_2) \cdot \prod_{i=1}^n e(u_i, S_{5,i})$ and $e(S_{5,i}, g) = e(X_i, S_2)$ for $i = 1, \ldots, n$.

The security of this simplified scheme can only be proven if, when answering certification queries, the simulator can control the private keys $(x_1, \ldots, x_n)$ and force them to be random values of its choice. To allow the simulator to sign arbitrary public keys without knowing the private keys, we modify the scheme so that the signer rather signs commitments (calculated using our structure-preserving trapdoor commitment) to public key elements $X_1, \ldots, X_n$. In the security proof, the simulator first generates a signature on n fake commitments $\mathbf{C}_i = (C_{i,1}, C_{i,2}, C_{i,3})$ that are all generated in such a way that it knows $\log_g(C_{i,j})$ for $i = 1, \ldots, n$ and $j = 1, 2, 3$. Using the trapdoor of the commitment scheme, it can then open $\mathbf{C}_i$ to any arbitrary $X_i \in \mathbb{G}$ without knowing $\log_g(X_i)$.

This use of the trapdoor commitment is reminiscent of a technique (notably used in [89]) to construct signature schemes in the standard model using chameleon hash functions [162]: the simulator first signs messages of its choice using a basic signature scheme and then "equivocates" the chameleon hashes to make them correspond to adversarially-chosen messages.

Keygen(pp, n)**:** given common public parameters $\mathsf{pp} = \{g, \mathbb{G}, \mathbb{G}_T\}$, select $u, u_0 \stackrel{\$}{\leftarrow} \mathbb{G}$ as well as $\alpha, \omega \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$ and set $A = e(g,g)^\alpha$, $\Omega = g^\omega$. Then, pick $\beta_{i,1}, \beta_{i,2}, \beta_{i,3} \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$ and define

$$\overline{u}_i = (u_{i,1}, u_{i,2}, u_{i,3}) = (g^{\beta_{i,1}}, g^{\beta_{i,2}}, g^{\beta_{i,3}})$$

for $i = 1, \ldots, n$. Choose $f, f_1, f_2, f_{3,1}, f_{3,2}, f_{3,3} \stackrel{\$}{\leftarrow} \mathbb{G}$ that define a commitment key consisting of vectors $\mathbf{f_1} = (f_1, 1, f)$, $\mathbf{f_2} = (1, f_2, f)$ and $\mathbf{f_3} = (f_{3,1}, f_{3,2}, f_{3,3})$. Define the private key to be $SK = \big(\alpha, \omega, \{\overline{\beta}_i = (\beta_{i,1}, \beta_{i,2}, \beta_{i,3})\}_{i=1,\ldots,n}\big)$ and the public key as

$$PK \quad = \quad \Big(\mathbf{f} = (\mathbf{f_1}, \mathbf{f_2}, \mathbf{f_3}),\ A = e(g,g)^\alpha,\ \Omega = g^\omega,\ u,\ u_0,\ \{\overline{u}_i\}_{i=1,\ldots,n}\Big).$$

Sign(pp, SK, M)**:** parse SK as $\big(\alpha, \omega, \{\overline{\beta}_i\}_{i=1,\ldots,n}\big)$, M as $(X_1, \ldots, X_n)$ and do the following.

1. For each $i \in \{1, \ldots, n\}$, pick $\phi_{i,1}, \phi_{i,2}, \phi_{i,3} \overset{\$}{\leftarrow} \mathbb{Z}_p^*$ and compute a commitment

$$C_i = (C_{i,1}, C_{i,2}, C_{i,3}) = (f_1^{\phi_{i,1}} \cdot f_{3,1}^{\phi_{i,3}},\ f_2^{\phi_{i,2}} \cdot f_{3,2}^{\phi_{i,3}},\ X_i \cdot f^{\phi_{i,1}+\phi_{i,2}} \cdot f_{3,3}^{\phi_{i,3}})$$

   and the matching de-commitment $(D_{i,1}, D_{i,2}, D_{i,3}) = (f^{\phi_{i,1}}, f^{\phi_{i,2}}, f^{\phi_{i,3}})$.

2. Choose $c_{\mathcal{ID}} \overset{\$}{\leftarrow} \mathbb{Z}_p^*$ and compute $S_1 = (g^{\alpha})^{1/(\omega + c_{\mathcal{ID}})}$, $S_2 = g^{c_{\mathcal{ID}}}$, $S_3 = u^{c_{\mathcal{ID}}}$ as well as

$$\begin{aligned} S_4 &= \Big(u_0 \cdot \prod_{i=1}^{n} (C_{i,1}^{\beta_{i,1}} \cdot C_{i,2}^{\beta_{i,2}} \cdot C_{i,3}^{\beta_{i,3}})\Big)^{c_{\mathcal{ID}}} \\ S_5 &= \{(S_{5,i,1}, S_{5,i,2}, S_{5,i,3})\}_{i=1,\ldots,n} = \{(C_{i,1}^{c_{\mathcal{ID}}}, C_{i,2}^{c_{\mathcal{ID}}}, C_{i,3}^{c_{\mathcal{ID}}})\}_{i=1,\ldots,n} \end{aligned}$$

Return $\mathsf{cert}_M = \Big(\{(C_{i,1}, C_{i,2}, C_{i,3}), (D_{i,1}, D_{i,2}, D_{i,3})\}_{i=1,\ldots,n}, S_1, S_2, S_3, S_4, S_5\Big)$.

Verify$(\mathsf{pp}, PK, M, \mathsf{cert}_M)$**:** parse M as $(X_1, \ldots, X_n)$ and cert_M as above. Return 1 if, for indices $i = 1, \ldots, n$, it holds that $X_i \in \mathbb{G}$ and

$$e(C_{i,1}, f) = e(f_1, D_{i,1}) \cdot e(f_{3,1}, D_{i,3}) \tag{2.1}$$

$$e(C_{i,2}, f) = e(f_2, D_{i,2}) \cdot e(f_{3,2}, D_{i,3}) \tag{2.2}$$

$$e(C_{i,3}, f) = e(X_i \cdot D_{i,1} \cdot D_{i,2}, f) \cdot e(f_{3,3}, D_{i,3}), \tag{2.3}$$

and if the following checks are also satisfied. Otherwise, return 0.

$$e(S_1, \Omega \cdot S_2) = A \tag{2.4}$$

$$e(S_2, u) = e(g, S_3) \tag{2.5}$$

$$e(S_4, g) = e(u_0, S_2) \cdot \prod_{i=1}^{n} \big(e(u_{i,1}, S_{5,i,1}) \cdot e(u_{i,2}, S_{5,i,2}) \cdot e(u_{i,3}, S_{5,i,3})\big), \tag{2.6}$$

$$e(S_{5,i,j}, g) = e(C_{i,j}, S_2) \qquad \text{for } i = 1, \ldots, n,\ j = 1, 2, 3 \tag{2.7}$$

A signature on $(X_1, \ldots, X_n) \in \mathbb{G}^n$ is comprised of $9n + 4$ group elements. Subsequently to our work, Abe *et al.* [6, 4] showed how to sign messages in $\mathbb{G}^n$ using $O(1)$ group elements.

We note that the scheme is not structure-preserving in the strict sense since the public key component $A = e(g, g)^a$ lives in the group $\mathbb{G}_T$. However, everything goes through if $A = e(g, g)^a$ is replaced by a pair of public group elements $(A_1, A_2) \in \mathbb{G}^2$ such that $e(A_1, A_2) = e(g, g)^a$.

Regarding the security of the scheme, the following theorem is proved in [81].

Theorem 1 ([81])**.** *The scheme is secure under chosen-message attacks if the HSDH, FlexDH and SDP problems are all hard in* $\mathbb{G}$.

The scheme can also be used to construct non-frameable group signatures that are secure in the concurrent join model of [157] without resorting to random oracles. To the best of our knowledge, before 2009, the Kiayias-Yung construction [157] was the only scalable group signature where joining supports concurrency at both ends while requiring the smallest amount of interaction. In the standard model, our signature scheme thus provided the first[2] way to achieve the same result. In this case, we have $n = 1$ (since prospective group members only need to certify one group element if non-frameability is ensured by signing messages using Boneh-Boyen signatures [42] in the same way as in Groth's group signature [134]) so that membership certificates comprise 13 group elements and their shape is fully compatible with GS proofs.

2.1.3 A Group Encryption Scheme with Non-Interactive Proofs

In [81], we built a non-interactive GE scheme for the Diffie-Hellman relation $\mathcal{R} = \{(X,Y), W\}$ where $e(g,W) = e(X,Y)$, for which the keys are $\mathsf{pk}_{\mathcal{R}} = \{\mathbb{G}, \mathbb{G}_T, g\}$ and $\mathsf{sk}_{\mathcal{R}} = \varepsilon$. While our example is for the Diffie-Hellman relation, it can be easily generalized to any relation that can be expressed in terms of pairing-product equations for which NIZK proofs are available.

The construction slightly departs from the modular design of [156] in that commitments to the receiver's public key and certificate are part of the proof (instead of the ciphertext), which simplifies the proof of message-security. The security of the scheme eventually relies on the HSDH, FlexDH and DLIN assumptions. All security proofs are available in the full version of [81].

The group manager uses a key pair for our structure-preserving signature of Section 2.1.2 to sign public keys of the DLIN-based version [143, 235] of the Cramer-Shoup cryptosystem [88]. In the latter system, if we assume public generators g_1, g_2, g that are parts of public parameters, each receiver's public key is made of $n = 6$ group elements

$$\begin{array}{rclccc} X_1 &=& g_1^{x_1} g^{x} & X_3 = g_1^{x_3} g^{y} & X_5 = g_1^{x_5} g^{z} \\ X_2 &=& g_2^{x_2} g^{x} & X_4 = g_2^{x_4} g^{y} & X_6 = g_2^{x_6} g^{z}. \end{array}$$

To encrypt a plaintext $m \in \mathbb{G}$ under the label[3] L (see [238] for a definition of encryption schemes with labels), the sender picks $r, s \xleftarrow{\$} \mathbb{Z}_p^*$ and computes

$$\psi_{\mathsf{CS}} = (U_1, U_2, U_3, U_4, U_5) = \Big(g_1^r,\ g_2^s,\ g^{r+s},\ m \cdot X_5^r X_6^s,\ (X_1 X_3^\alpha)^r \cdot (X_2 X_4^\alpha)^s\Big),$$

where $\alpha = H(U_1, U_2, U_3, U_4, L) \in \mathbb{Z}_p^*$ is a collision-resistant hash[4]. Given (ψ_{CS}, L), the receiver computes α. He returns $\perp$ if $U_5 \neq U_1^{x_1+\alpha x_3} U_2^{x_2+\alpha x_4} U_3^{x+\alpha y}$ and $m = U_4/(U_1^{x_5} U_2^{x_6} U_3^{z})$ otherwise.

Our GE scheme goes as follows.

[2] Non-frameable group signatures described in [95, 54] achieve concurrent security by having the prospective user generate an extractable commitment to some secret exponent (which the simulator can extract without rewinding using the trapdoor of the commitment) and prove that the committed value is the discrete log. of a public value. In the standard model, this technique requires interaction and the proof should be simulatable in zero-knowledge when proving security against framing attacks. Another technique [113] requires users to prove knowledge of their secret exponent using Groth-Sahai non-interactive proofs. It is nevertheless space-demanding as each bit of committed exponent requires its own extractable GS commitment.

[3] A label is basically a set of public data that is bound to the ciphertext in a non-malleable manner.

[4] The proof of CCA2-security [88, 235] only requires a universal one-way hash function (UOWHF) [207] but collision-resistance is required when the scheme uses labels.

$\mathsf{SETUP}_{\mathsf{init}}(\lambda)$: choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of order $p > 2^\lambda$, $g \xleftarrow{\$} \mathbb{G}$ and $g_1 = g^{\alpha_1}, g_2 = g^{\alpha_2}$ with $\alpha_1, \alpha_2 \xleftarrow{\$} \mathbb{Z}_p^*$. Define $\mathbf{g_1} = (g_1, 1, g)$, $\mathbf{g_2} = (1, g_2, g)$ and $\mathbf{g_3} = \mathbf{g_1}^{\xi_1} \cdot \mathbf{g_2}^{\xi_2}$ with $\xi_1, \xi_2 \xleftarrow{\$} \mathbb{Z}_p^*$, which form a CRS $\mathbf{g} = (\mathbf{g_1}, \mathbf{g_2}, \mathbf{g_3})$ for the perfect soundness setting. Select a strongly unforgeable (as defined in [12]) one time signature scheme $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ and a random member $H : \{0,1\}^* \to \mathbb{Z}_p$ of a collision-resistant hash family. Public parameters consists of $\mathsf{param} = \{\lambda, \mathbb{G}, \mathbb{G}_T, g, \mathbf{g}, \Sigma, H\}$.

$\mathsf{SETUP}_{\mathsf{GM}}(\mathsf{params})$: runs the setup algorithm of the certification scheme described in section 2.1.2 with $n = 6$. The obtained public key consists of

$$\mathsf{pk}_{\mathsf{GM}} = \Big(\mathbf{f},\ A = e(g,g)^\alpha,\ \Omega = g^\omega,\ u,\ u_0,\ \{\overline{u}_i\}_{i=1,\ldots,6}\Big)$$

and the matching private key is $\mathsf{sk}_{\mathsf{GM}} = \big(\alpha, \omega, \{\overline{\beta}_i = (\beta_{i,1}, \beta_{i,2}, \beta_{i,3})\}_{i=1,\ldots,6}\big)$.

$\mathsf{SETUP}_{\mathsf{OA}}(\mathsf{params})$: generates $\mathsf{pk}_{\mathsf{OA}} = (Y_1, Y_2, Y_3, Y_4) = (g^{y_1}, g^{y_2}, g^{y_3}, g^{y_4})$, as a public key for Kiltz's tag-based encryption (TBE) scheme [160], and the corresponding private key as $\mathsf{sk}_{\mathsf{OA}} = (y_1, y_2, y_3, y_4)$.

JOIN: the user sends a linear Cramer-Shoup public key $\mathsf{pk} = (X_1, \ldots, X_6) \in \mathbb{G}^6$ to the GM and obtains a certificate

$$\mathsf{cert}_{\mathsf{pk}} = \big(\{(C_{i,1}, C_{i,2}, C_{i,3}), (D_{i,1}, D_{i,2}, D_{i,3})\}_{i=1,\ldots,6}, S_1, S_2, S_3, S_4, S_5\big).$$

$\mathsf{ENC}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}_{\mathsf{pk}}, W, L)$: to encrypt $W \in \mathbb{G}$ such that $((X, Y), W) \in \mathcal{R}$ (for public elements $X, Y \in \mathbb{G}$), parse $\mathsf{pk}_{\mathsf{GM}}$, $\mathsf{pk}_{\mathsf{OA}}$ and pk as above and do the following.

1. Generate a one-time signature key pair $(\mathsf{SK}, \mathsf{VK}) \leftarrow \mathcal{G}(\lambda)$.
2. Choose $r, s \xleftarrow{\$} \mathbb{Z}_p^*$ and compute a linear CS encryption of W, the result of which is denoted by ψ_{CS}, under the label $L_1 = L||\mathsf{VK}$ (and using the collision-resistant hash function specified by params).
3. For $i = 1, \ldots, 6$, choose $w_{i,1}, w_{i,2} \xleftarrow{\$} \mathbb{Z}_p^*$ and encrypt X_i under $\mathsf{pk}_{\mathsf{OA}}$ using Kiltz's TBE scheme [160] with the tag VK. Let

$$\psi_{\mathsf{K}_i} = (Y_1^{w_{i,1}}, Y_2^{w_{i,2}}, (g^{\mathsf{VK}} Y_3)^{w_{i,1}}, (g^{\mathsf{VK}} Y_4)^{w_{i,2}}, X_i \cdot g^{w_{i,1}+w_{i,2}})$$

be the ciphertexts.

4. Set the GE ciphertext ψ as $\psi = \mathsf{VK}||\psi_{\mathsf{CS}}||\psi_{\mathsf{K}_1}||\cdots||\psi_{\mathsf{K}_6}||\sigma$ where σ is a one-time signature obtained as $\sigma = \mathcal{S}(\mathsf{sk}, (\psi_{\mathsf{CS}}||\psi_{\mathsf{K}_1}||\cdots||\psi_{\mathsf{K}_6}||L))$.

Return (ψ, L) and $coins_\psi$ consist of $\{(w_{i,1}, w_{i,2})\}_{i=1,\ldots,6}$, (r, s). If the one-time signature of [133] is used, VK and σ take 3 and 2 group elements, respectively, so that ψ comprises 40 group elements.

$\mathcal{P}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}_{\mathsf{pk}}, (X, Y), W, \psi, L, coins_\psi)$: parse $\mathsf{pk}_{\mathsf{GM}}$, $\mathsf{pk}_{\mathsf{OA}}$, pk and ψ as above. Conduct the following steps.

1. Generate commitments (as explained in section 1.3) to the $9n+4=58$ group elements that $\mathsf{cert}_{\mathsf{pk}}$ consists of. The resulting overall commitment $com_{\mathsf{cert}_{\mathsf{pk}}}$ contains 184 group elements.
2. Generate GS commitments to the public key elements $\mathsf{pk}=(X_1,\ldots,X_6)$ and obtain the set $com_{\mathsf{pk}}=\{com_{X_i}\}_{i=1,\ldots,6}$, which consists of 18 group elements.
3. Generate a proof $\pi_{\mathsf{cert}_{\mathsf{pk}}}$ that $com_{\mathsf{cert}_{\mathsf{pk}}}$ is a commitment to a valid certificate for the public key contained in com_{pk}. For each $i=1,\ldots,6$, relations (2.1)-(2.3) cost 9 elements to prove (and thus 54 elements altogether). The quadratic equation (2.4) takes 9 elements and linear ones (2.5)-(2.6) both require 3 elements. Finally, (2.7) is a set of 18 linear equations which demand 54 elements altogether. The whole proof $\pi_{\mathsf{cert}_{\mathsf{pk}}}$ thus takes 123 group elements.
4. For $i=1,\ldots,6$, generate a NIZK proof $\pi_{eq\text{-}key,i}$ that com_{X_i} (which is part of com_{pk}) and ψ_{K_i} are encryptions of the same X_i. If ψ_{K_i} comprises
$$(V_{i,1},V_{i,2},V_{i,5})=(Y_1^{w_{i,1}},Y_2^{w_{i,2}},X_i\cdot g^{w_{i,1}+w_{i,2}})$$
and com_{X_i} is parsed as $(c_{X_{i1}},c_{X_{i2}},c_{X_{i3}})=(g_1^{\theta_{i1}}\cdot g_{3,1}^{\theta_{i3}},\ g_2^{\theta_{i2}}\cdot g_{3,2}^{\theta_{i3}},\ X_i\cdot g^{\theta_{i1}+\theta_{i2}}\cdot g_{3,3}^{\theta_{i3}})$, where $w_{i,1},w_{i,2}\in coins_\psi$, $\theta_{i1},\theta_{i2},\theta_{i3}\in\mathbb{Z}_p^*$ and $\mathbf{g_3}=(g_{3,1},g_{3,2},g_{3,3})$, this amounts to prove knowledge of values $w_{i,1},w_{i,2},\theta_{i1},\theta_{i2},\theta_{i3}\in\mathbb{Z}_p^*$ such that
$$\Big(\frac{V_{i,1}}{c_{X_{i1}}},\frac{V_{i,2}}{c_{X_{i2}}},\frac{V_{i,3}}{c_{X_{i3}}}\Big)=\big(Y_1^{w_{i,1}}\cdot g_1^{-\theta_{i1}}\cdot g_{3,1}^{-\theta_{i3}},\ Y_2^{w_{i,2}}\cdot g_2^{-\theta_{i2}}\cdot g_{3,2}^{-\theta_{i3}},\ g^{w_{i,1}+w_{i,2}-\theta_{i1}-\theta_{i2}}\cdot g_{3,3}^{-\theta_{i3}}\big).$$
Committing to the encryption exponents $w_{i,1},w_{i,2},\theta_{i1},\theta_{i2},\theta_{i3}$ introduces 90 group elements whereas the above relations only require two elements each. Overall, proof elements $\pi_{eq\text{-}key,1},\ldots,\pi_{eq\text{-}key,6}$ incur 126 elements.
5. Generate a NIZK proof $\pi_{val\text{-}enc}$ that $\psi_{\mathsf{CS}}=(U_1,U_2,U_3,U_4,U_5)$ is a valid CS encryption. This requires to commit to underlying encryption exponents $r,s\in coins_\psi$ and prove that $U_1=g_1^r$, $U_2=g_2^s$, $U_3=g^{r+s}$ (which only takes 3 times 2 elements as base elements are public) and $U_5=(X_1X_3^\alpha)^r\cdot(X_2X_4^\alpha)^s$ (which takes 9 elements since base elements are themselves variables). Including commitments com_r and com_s to exponents r and s, $\pi_{val\text{-}enc}$ demands 21 group elements overall.
6. Generate a NIZK proof $\pi_{\mathcal{R}}$ that the ciphertext ψ_{CS} encrypts a group element $W\in\mathbb{G}$ such that $((X,Y),W)\in\mathcal{R}$. To this end, generate a commitment
$$com_W=(c_{W,1},c_{W,2},c_{W,3})=(g_1^{\theta_1}\cdot g_{3,1}^{\theta_3},\ g_2^{\theta_2}\cdot g_{3,2}^{\theta_3},\ W\cdot g^{\theta_1+\theta_2}\cdot g_{3,3}^{\theta_3})$$
and prove that the underlying W is the same as the one for which $U_4=W\cdot X_5^r\cdot X_6^s$ in ψ_{CS}. In other words, prove knowledge of exponents $r,s,\theta_1,\theta_2,\theta_3$ such that
$$\Big(\frac{U_1}{c_{W,1}},\frac{U_2}{c_{W,2}},\frac{U_4}{c_{W,3}}\Big)=\big(g_1^{r-\theta_1}\cdot g_{3,1}^{-\theta_3},\ g_2^{s-\theta_2}\cdot g_{3,2}^{-\theta_3},\ g^{-\theta_1-\theta_2}\cdot g_{3,3}^{-\theta_3}\cdot X_5^r\cdot X_6^s\big).\qquad(2.8)$$
Commitments to r,s are already part of $\pi_{val\text{-}enc}$. Committing to $\theta_1,\theta_2,\theta_3$ takes 9 elements. Proving the first two relations of (2.8) requires 4 elements whereas the third one is quadratic and its proof is 9 elements. Proving the linear pairing-product relation $e(g,W)=e(X,Y)$ in NIZK[5] demands 9 elements. Since $\pi_{\mathcal{R}}$ includes com_W, it entails a total of 34 elements.

[5] It requires to introduce an auxiliary variable $\mathcal{X}$ and prove that $e(g,\mathcal{W})=e(\mathcal{X},Y)$ and $\mathcal{X}=X$, for variables $\mathcal{W},\mathcal{X}$ and constants g,X,Y. The two proofs take 3 elements each and 3 elements are needed to commit to $\mathcal{X}$.

The entire proof $\pi_\psi = com_{\mathsf{cert}_{\mathsf{pk}}}||com_{\mathsf{pk}}||\pi_{\mathsf{cert}_{\mathsf{pk}}}||\pi_{eq\text{-}key,1}||\cdots||\pi_{eq\text{-}key,6}||\pi_{val\text{-}enc}||\pi_{\mathcal{R}}$ eventually takes 516 elements.

$\mathcal{V}(\mathsf{params}, \psi, L, \pi_\psi, \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}})$: parse $\mathsf{pk}_{\mathsf{GM}}$, $\mathsf{pk}_{\mathsf{OA}}$, pk, ψ and π_ψ as above. Return 1 if and only if $\mathcal{V}(\mathsf{VK}, \sigma, (\psi_{\mathsf{CS}}||\psi_{\mathsf{K}_1}||\cdots||\psi_{\mathsf{K}_6}||L)) = 1$, all proofs verify and if $\psi_{\mathsf{K}_1}, \ldots, \psi_{\mathsf{K}_6}$ are all valid tag-based encryptions w.r.t. the tag VK.

$\mathsf{DEC}(\mathsf{sk}, \psi, L)$: parse the ciphertext ψ as $\mathsf{VK}||\psi_{\mathsf{CS}}||\psi_{\mathsf{K}_1}||\cdots||\psi_{\mathsf{K}_6}||\sigma$. Return $\perp$ in the event that $\mathcal{V}(\mathsf{VK}, \sigma, (\psi_{\mathsf{CS}}||\psi_{\mathsf{K}_1}||\cdots||\psi_{\mathsf{K}_6}||L)) = 0$. Otherwise, use sk to decrypt (ψ_{CS}, L).

$\mathsf{OPEN}(\mathsf{sk}_{\mathsf{OA}}, \psi, L)$: parse ψ as $\mathsf{VK}||\psi_{\mathsf{CS}}||\psi_{\mathsf{K}_1}||\cdots||\psi_{\mathsf{K}_6}||\sigma$. Return $\perp$ if $\psi_{\mathsf{K}_1}, \ldots, \psi_{\mathsf{K}_6}$ are not all valid TBE ciphertexts w.r.t. the tag VK or if $\mathcal{V}(\mathsf{VK}, \sigma, (\psi_{\mathsf{CS}}||\psi_{\mathsf{K}_1}||\cdots||\psi_{\mathsf{K}_6}||L)) = 0$. Otherwise, decrypt $\psi_{\mathsf{K}_1}, \ldots, \psi_{\mathsf{K}_6}$ using $\mathsf{sk}_{\mathsf{OA}}$ and return the resulting $\mathsf{pk} = (X_1, \ldots, X_6)$.

The following security result was proved in [81].

Theorem 2 ([81]). *The above group encryption system provides message privacy, anonymity and soundness assuming that H is a collision-resistant hash function and that the HSDH, FlexDH, and DLIN problems are all hard in $\mathbb{G}$.*

From an efficiency standpoint, the length of ciphertexts is about 4.5 kB in an implementation using symmetric pairings with a 512-bit group order. Moreover, our proofs only require 32.250 kB. This is significantly cheaper than in the original GE scheme [156] where, for 1024-bit RSA moduli, interactive proofs reach a communication cost of 70 kB to achieve a 2^{-50} knowledge error.

Of course, the above construction can be made significantly more efficient if our structure-preserving signature is replaced by the construction of Abe *et al.* [6], which was recalled in Section 1.5. In [176], we used the latter SPS system to build a group encryption scheme where, as in traceable signatures [155], the tracing authority can release a user-specific trapdoor that allows tracing all ciphertexts encrypted for a given user.

2.2 Group Signatures with Efficient Revocation in the Standard Model

Group signatures are a central cryptographic primitive, suggested by Chaum and van Heyst [85], which allows members of a population of users managed by some authority to sign messages in the name of the group while hiding their identity. At the same time, a tracing authority is capable of identifying the signer if necessary. A crucial problem is the revocation of the anonymous signing capability of users when they are banned from or intentionally leave the group.

2.2.1 Related Work

GROUP SIGNATURES. The first efficient and provably coalition-resistant group signature dates back to the work of Ateniese, Camenisch, Joye and Tsudik [15]. By the time their scheme appeared, the security of the primitive was not appropriately formalized yet. Suitable security definitions remained lacking until the work of Bellare, Micciancio and Warinschi [31] (BMW) who captured all the requirements of group signatures in three properties.

In (a variant of) this model, Boneh, Boyen and Shacham [44] obtained very short signatures using the random oracle methodology [32].

The BMW model assumes static groups where no new member can be introduced after the setup phase. The setting of dynamically changing groups was analyzed later on by Bellare-Shi-Zhang [33] and, independently, by Kiayias and Yung [158]. In the models of [33, 158], constructions featuring relatively short signatures were proposed in [210, 95]. A construction in the standard model was also suggested by Ateniese *et al.* [14] under interactive assumptions. At the same time, Boyen and Waters gave a different solution [55] without random oracles using more standard assumptions. By improving upon their own scheme, they managed [56] to obtain signatures of constant size. Their constructions [55, 56] were both presented in the BMW model [31] and provide anonymity in the absence of signature opening oracle. In the dynamic model [33], Groth [133] showed a system in the standard model with $O(1)$-size signatures but, due to very large hidden constants, his scheme was mostly a feasibility result. Later on, Groth came up with an efficient realization [134] (and signatures of about 50 group elements) with the strongest anonymity level.

REVOCATION. As in ordinary PKIs, where certificate revocation is a critical issue, membership revocation is a complex problem that has been extensively studied [57, 17, 68, 52] in the last decade. Generating a new group public key and distributing new signing keys to unrevoked members is a simple solution. In large groups, it is impractical to update the public key and provide members with new keys after they joined the group. Bresson and Stern suggested a different approach [57] consisting of having the signer prove that his membership certificate does not belong to a list of revoked certificates. Unfortunately, the length of signatures grows with the number of revoked members. In forward-secure group signatures, Song [240] chose a different way to handle revocation but verification takes linear time in the number of excluded users.

Camenisch and Lysyanskaya [68] proposed an elegant method using accumulators[6] [34]. Their technique, also used in [243, 66], allows revoking members while keeping $O(1)$ costs for signing and verifying. The downside of this approach is its history-dependence: it requires users to follow the dynamic evolution of the group and keep track of all changes: each revocation incurs a modification of the accumulator value, so that unrevoked users have to upgrade their membership certificate before signing new messages. In the worst case, this may require up to $O(r)$ exponentiations, if r is the number of revoked users.

Another drawback of accumulator-based approaches is their limited applicability in the standard model. Indeed, for compatibility reasons with the central tool of Groth-Sahai proofs, pairing-based accumulators are the only suitable candidates. However, in known pairing-based accumulators [209, 66], public keys have linear size in the maximal number of accumulations, which would result in linear-size group public keys in immediate implementations. To address this concern in delegatable anonymous credentials, Acar and Nguyen [8] chose to sacrifice the constant size of proofs of non-membership but, in group signatures, this would prevent signatures from having constant size. Boneh, Boyen and Shacham [44] managed to avoid linear dependencies in a revocation mechanism along the lines of [68]. Unfortunately, their technique does not seem to readily interact[7] with Groth-Sahai proofs

[6] An accumulator is a kind of "hash" function mapping a set of values to a short, constant-size string while allowing to efficiently prove that a specific value was accumulated.

[7] In [44], signing keys consist of pairs $(g^{1/(\omega+s)}, s) \in \mathbb{G} \times \mathbb{Z}_p$, where $\omega \in \mathbb{Z}_p$ is the secret key of the group manager, and the revocation method relies on the availability of the exponent $s \in \mathbb{Z}_p$. In the standard model,

[138] so as to work in the standard model. Moreover, like the Camenisch-Lysyanskaya technique [68], the Boneh-Boyen-Shacham method may require up to $O(r)$ exponentiations to update unrevoked users' private keys if r is the cardinality of the processed revocation list.

In [58], Brickell considered the notion of *verifier-local revocation* group signatures, for which formal definitions were given by Boneh and Shacham [52] and other extensions were proposed in [203, 251, 185]. In this approach, revocation messages are only sent to verifiers and the signing algorithm is completely independent of the number of revocations. Verifiers take as additional input a revocation list (RL), maintained by the group manager, and have to perform a revocation test for each RL entry in order to be convinced that signatures were not issued by a revoked member (a similar revocation mechanism is used in [59]). The verification cost is thus inevitably linear in the number of expelled users.

In 2009, Nakanishi, Fuji, Hira and Funabiki [202] came up with a revocable group signature with constant complexities for signing/verifying. At the same time, group members never have to update their keys. On the other hand, their proposal suffers from linear-size group public keys in the maximal number N of users, although a variant reduces the group public key size to $O(N^{1/2})$.

In anonymous credentials, Tsang *et al.* [241, 242] showed how to prevent users from anonymously authenticating themselves without compromising their anonymity or involving a trusted third party. Their schemes either rely on accumulators (which may be problematic in our setting) or have linear proving complexity in the number of revocations. Camenisch, Kohlweiss and Soriente [67] dealt with revocations in anonymous credentials by periodically updating users credentials in which a specific attribute indicates a validity period. In group signatures, their technique would place an important burden on the group manager who would have to generate updates for each unrevoked individual credential.

2.2.2 Our Results

For various reasons, none of the previously mentioned constructions conveniently supports large groups, especially if we restrict ourselves to constructions without random oracles.

Together with Moti Yung and Thomas Peters [180], we described a novel revocation mechanism, borrowed from the literature on broadcast encryption, which is truly scalable and well-suited to constructions in the standard model. Using the Subset Cover framework of Naor, Naor and Lotspiech [205] (NNL), we provided two distinct constructions [180, 179] of history-independent revocable group signatures in the standard model. Our technique [180] blends well with structure-preserving signatures and Groth-Sahai proofs.

Constructions with polylog-size private keys

As in the NNL Subset Cover framework [205], our first revocable group signature assigns each group member to a leaf of a binary tree and, at any time, the set $\{1,\ldots,N\}\backslash\mathcal{R}$ of unrevoked group members is partitioned into a collection $S_1,\ldots,S_m$ of disjoint subsets of leaves, for some $m \in \mathbb{N}$. Each unrevoked member should belong to exactly one subset S_i in the cover of authorized leafs determined by the group manager. In order to sign a message, an authorized member thus has to demonstrate that he is not revoked by proving his membership of one of the subsets S_i without revealing which one. In its best tradeoff, our first

the Groth-Sahai techniques would require to turn the membership certificates into triples $(g^{1/(\omega+s)}, g^s, u^s)$, for some $u \in \mathbb{G}$ (as in [56]), which is not compatible with the revocation mechanism.

construction [180] builds on the public-key variant, due to Dodis and Fazio [99], of the Subset Difference (SD) method [205], where unrevoked group members $\{1, \ldots, N\} \backslash \mathcal{R} = \bigcup_{i=1}^{m} S_i$ are partitioned into a collection of $m = O(|\mathcal{R}|)$ subsets, each of which is the difference between two sub-trees.

Like the Dodis-Fazio construction [99], our first group signature builds on hierarchical identity-based encryption (HIBE) and uses the property that, in the broadcast encryption system of [99], each ciphertext can be seen as a collection of $m = O(|\mathcal{R}|)$ HIBE ciphertexts (one for each subset S_i of the partition), which is turned into a revocation list. In short, our group signature can be seen as having authorized group members prove that they are not revoked by showing their ability to decrypt a HIBE ciphertext contained in the revocation list. Of course, for anonymity purposes, the signer should not reveal which HIBE ciphertext he is able to decrypt since it would leak information on his position in the tree. For this reason, the relevant entry of the revocation list only appears in committed form in the group signature. In order to prove that he is using a legal entry of the revocation list, the user generates a set membership proof [61] and proves knowledge of a signature from the group manager on the committed RL entry. It is worth noting that RLs are *not* part of the group public key: verifiers only need to know the number of the latest revocation epoch and they should not bother to read RLs entirely.

This method features constant signature size and verification time, $O(\log N)$-size group public keys, revocation lists of size $O(r)$ (as in standard PKIs and group signatures with verifier-local revocation) and membership certificates of size $O(\log^3 N)$. In a different tradeoff of the same high-level construction, we can reduce the private key size to $O(\log N)$ using the Complete Subtree method [205]. In this case, however, revocation lists are inflated by a factor of $O(\log N/r)$. While the Layered Subset Difference method [140] allows for noticeable improvements, the constructions of [180] still suffer from relatively large membership certificates. We remark, however, that some logarithmic dependency is expected when basing revocation on a tree-like NNL methodology.

For groups of N members, our first constructions thus feature constant-size signatures and verification time at the cost of membership certificates of size $O(\log^3 N)$ (or $O(\log^{2.5} N)$ using the Layered Subset Difference method). In many applications, this can become rather expensive even for moderately large groups: for example, using the Subset Difference method with $N = 1000 \approx 2^{10}$, users may have to privately store thousands of group elements. In order to be competitive with other group signatures in the standard model such as [134] and still be able to revoke members while keeping them "stateless", it is desirable to avoid this storage complexity.

Constructions with Short Private Keys

In our second main construction of revocable group signature [179], we managed to get rid of the polylogarithmic complexity in the private key size and obtained constant-size membership certificates while retaining the same complexities in other metrics. This improvement was achieved at the expense of relying on a somewhat stronger (but still falsifiable) hardness assumption in the security proofs.

Our improved construction [179] also builds on the NNL Subset Cover framework [205] to partition the subset of authorized users using the Subset Difference method. However, instead of relying on a broadcast encryption system, it leverages the properties of a special kind of commitment schemes introduced by Moti Yung and myself in 2010 [188]. These commitments yield private keys of *constant* size without degrading other performance criteria. This may sound somewhat surprising since, in the SD method, (poly)logarithmic complexities inherently seem inevitable in several metrics. Indeed, in the context of broadcast encryption [205], it requires private keys of size $O(\log^2 N)$ (and even $O(\log^3 N)$ in the public key setting [99] if the result of Boneh-Boyen-Goh [43] is used). Here, we reduce this overhead to a constant while the only dependency on N is a $O(\log N)$-size group public key.

Instead of relying on hierarchical identity-based encryption [45, 144, 123] as in the public-key variant [99] of NNL, our improved construction employs *concise* vector commitment schemes [188, 75], where each commitment can be opened w.r.t. individual coordinates in a space-efficient manner (namely, the size of a coordinate-wise opening does not depend on the length of the vector). These vector commitments interact nicely with the specific shape of subsets – as differences between two subtrees – in the SD method. Using them, we compactly encode as a vector the path from the user's leaf to the root. To provide evidence of their inclusion in one of the SD subsets, group members successively prove the equality and the inequality between two coordinates of their vector (i.e., two nodes of the path from their leaf to the root) and specific node labels indicated by an appropriate entry of the revocation list. This is where the position-wise openability of concise commitments is very handy.

The use of concise commitments allows making the most of the Subset Cover approach [180] by reducing the size of membership certificates to a small constant: at the cost of lengthening signatures by a small constant factor (roughly 1.5), we obtain membership certificates consisting of only 9 group elements and a small integer. For $N = 1000$, users' private keys are thus compressed by a multiplicative factor of several hundreds and this can only become more dramatic for larger groups. At the same time, our main scheme retains all the useful properties of [180]: like the construction of Nakanishi *et al.* [202], it does not require users to update their membership certificates at any time but, unlike [202], our group public key size is $O(\log N)$. Like the SD-based construction of [180], our improved system uses revocation lists of size $O(r)$, which is on par with Certificate Revocation Lists (CRLs) of standard PKIs.

Eventually, we thus obtain revocable group signatures that become competitive with the regular CRL approach in PKIs: signature generation and verification have constant cost, signatures and membership certificates being of $O(1)$-size while revocation lists have size $O(r)$. It is conceivable that our improved revocation technique can find applications beyond group signatures.

2.2.3 Definition of Group Signatures with Revocation

We consider group signature schemes that have their lifetime divided into revocation periods at the beginning of which group managers update their revocation lists. The syntax and the security model are built on those defined by Kiayias and Yung [158]. Like the Bellare-Shi-Zhang model [33], the Kiayias-Yung (KY) model assumes an interactive *join* protocol whereby a prospective user becomes a group member by interacting with the group manager. This protocol provides the user with a membership certificate and a membership secret.

Syntax. We denote by $N \in \mathsf{poly}(\lambda)$ the maximal number of group members. At the beginning of each revocation period t, the group manager publicizes an up-to-date revocation list RL_t and we denote by $\mathcal{R}_t \subset \{1, \ldots, N\}$ the corresponding set of revoked users (we assume that $\mathcal{R}_t$ is part of RL_t). A revocable group signature (R-GS) scheme consists of the following algorithms or protocols.

Setup(λ, N)**:** given a security parameter $\lambda \in \mathbb{N}$ and a maximal number of group members $N \in \mathbb{N}$, this algorithm (which is run by a trusted party) generates a group public key $\mathcal{Y}$, the group manager's private key $\mathcal{S}_{\mathsf{GM}}$ and the opening authority's private key $\mathcal{S}_{\mathsf{OA}}$. Keys $\mathcal{S}_{\mathsf{GM}}$ and $\mathcal{S}_{\mathsf{OA}}$ are given to the appropriate authority while $\mathcal{Y}$ is publicized. The algorithm also initializes a public state St comprising a set data structure $St_{\mathsf{users}} = \emptyset$ and a string data structure $St_{\mathsf{trans}} = \epsilon$, which are initially empty.

Join: is an interactive protocol between the group manager GM and a user $\mathcal{U}_i$ who becomes a group member. The protocol involves two interactive Turing machines $\mathsf{J}_{\mathsf{user}}$ and J_{GM} that both take $\mathcal{Y}$ as input. The execution ends with user $\mathcal{U}_i$ obtaining a membership secret sec_i, that no one else knows, and a membership certificate cert_i. If the protocol is successful, the GM updates the public state St by setting $St_{\mathsf{users}} := St_{\mathsf{users}} \cup \{i\}$ as well as $St_{\mathsf{trans}} := St_{\mathsf{trans}} || \langle i, \mathsf{transcript}_i \rangle$.

Revoke: is a (possibly randomized) algorithm allowing the GM to generate an updated revocation list RL_t for the new revocation period t. It takes as input a public key $\mathcal{Y}$ and a set $\mathcal{R}_t \subset St_{\mathsf{users}}$ that identifies the users to be revoked. It outputs an updated revocation list RL_t for period t.

Sign: given a revocation period t with its revocation list RL_t, a membership certificate cert_i, a membership secret sec_i and a message M, this algorithm outputs $\perp$ if $i \in \mathcal{R}_t$ and a signature σ otherwise.

Verify: given a signature σ, a revocation period t, the corresponding revocation list RL_t, a message M and a group public key $\mathcal{Y}$, this algorithm returns either 0 or 1.

Open: takes as input a message M, a valid signature σ w.r.t. $\mathcal{Y}$ for the indicated revocation period t, the opening authority's private key $\mathcal{S}_{\mathsf{OA}}$ and the public state St. It outputs $i \in St_{\mathsf{users}} \cup \{\perp\}$, which is the identity of a group member or a symbol indicating an opening failure.

In our extension of the Kiayias-Yung model [158], a R-GS scheme must satisfy three security notions.

The first one is called *security against misidentification attacks*. It requires that, even if the adversary can introduce and revoke users at will, it cannot produce a signature that traces outside the set of unrevoked adversarially-controlled users. As in ordinary group signatures, the notion of *security against framing attacks* captures that under no circumstances should an honest user be held accountable for messages that he did not sign, even if the whole system conspired against him. Finally, the notion of *anonymity* is also defined by granting the adversary access to a signature opening oracle as in the models of [33, 158].

These security properties are formalized using experiments which are described in the articles in appendices. In short, they can be outlined as follows.

In a misidentification attack, the adversary can corrupt the opening authority. Moreover, he can also introduce malicious users in the group and revoke users at any time. His purpose is to come up with a signature $\sigma^\star$ that verifies w.r.t. $RL_{t^\star}$, where $t^\star$ denotes the current revocation period. He is deemed successful if the produced signature $\sigma^\star$ does not open to any unrevoked adversarially-controlled. The definition extends the usual definition [158] in that $\mathcal{A}$ also wins if his forgery $\sigma^\star$ verifies w.r.t. $RL_{t^\star}$ but opens to an adversarially-controlled user that *was* revoked during the revocation period $t^\star$.

Framing attacks consider the situation where the entire system, including the group manager and the opening authority, is colluding against some honest user. The adversary can corrupt the group manager as well as the opening authority. He is also allowed to introduce honest group members, observe the system while these users sign messages and create dummy users. In addition, before the possible corruption of the group manager, the adversary can revoke group members at any time. As a potentially corrupted group manager, $\mathcal{A}$ is allowed to come up with his own revocation list $RL_{t^\star}$ at the end of the game. We assume that anyone can publicly verify that $RL_{t^\star}$ is correctly formed so that the adversary does not come up with an ill-formed revocation list.

The notion of anonymity is formalized by means of a game involving a two-stage adversary. The first stage allows the adversary $\mathcal{A}$ to open arbitrary signatures by probing a signature opening oracle. When this stage ends, $\mathcal{A}$ chooses a message-period pair $(M^\star, t^\star)$ as well as two pairs $(\mathsf{sec}_0^\star, \mathsf{cert}_0^\star)$, $(\mathsf{sec}_1^\star, \mathsf{cert}_1^\star)$, consisting of a valid membership certificate and a corresponding membership secret. Then, the challenger flips a coin $d \leftarrow \{0,1\}$ and computes a challenge signature $\sigma^\star$ using $(\mathsf{sec}_d^\star, \mathsf{cert}_d^\star)$. The adversary is given $\sigma^\star$ with the task of eventually guessing the bit $d \in \{0,1\}$. Before doing so, he/she is allowed further oracle queries throughout the second stage, called guess stage, but is restricted not to query the opening oracle for $(M^\star, \sigma^\star, t^\star)$.

2.2.4 Our Construction with Short Private Keys

Our construction [179] with short private keys relies on concise vector commitment schemes, where commitments can be opened with a short de-commitment string for each individual coordinate. Such commitments based on ideas from [49, 66] were described by Libert and Yung [188] and, under weaker assumptions, by Catalano and Fiore [75]. In [188], the commitment key is $ck = (g, g_1, \ldots, g_\ell, g_{\ell+2}, \ldots, g_{2\ell}) \in \mathbb{G}^{2\ell}$, where $g_i = g^{(\alpha^i)}$ for each i. The trapdoor of the commitment is $g_{\ell+1}$, which does not appear in ck. To commit to a vector $(m_1, \ldots, m_\ell)$, the committer picks $r \overset{\$}{\leftarrow} \mathbb{Z}_p$ and computes $C = g^r \cdot \prod_{\kappa=1}^{\ell} g_{\ell+1-\kappa}^{m_\kappa}$. A single group element $W_i = g_i^r \cdot \prod_{\kappa=1,\kappa\neq i}^{\ell} g_{\ell+1-\kappa+i}^{m_\kappa}$ provides evidence that m_i is the i-th component of the vector as it satisfies the relation $e(g_i, C) = e(g, W_i) \cdot e(g_1, g_\ell)^{m_i}$. The infeasibility of opening a commitment to two distinct messages for some coordinate i relies on the ℓ-DHE assumption. For our purposes, we only rely on the position-wise binding property of vector commitments and do not need them to be hiding. The randomizer r will thus be removed from of C.

Intuition

The number of users is assumed to be $N = 2^{\ell-1} \in \mathsf{poly}(\lambda)$, for some integer ℓ, so that each group member is assigned to a leaf of the tree. Each node is assigned a unique identifier. For simplicity, the root is identified by $\mathcal{ID}(\epsilon) = 1$ and, for each other node x, we define the identifier $\mathcal{ID}(x) \in \{1, \ldots, 2N-1\}$ to be $\mathcal{ID}(x) = 2 \cdot \mathcal{ID}(\mathsf{parent}(x)) + b$, where $\mathsf{parent}(x)$ denotes x's father in the tree and $b = 0$ (resp. $b = 1$) if x is the left (resp. right) child of its father. The root of the tree is assigned the identifier $\mathcal{ID}(\epsilon) = 1$.

At the beginning of each revocation period t, the GM generates an up-to-date revocation list RL_t containing one entry for each generic subset $S_{k_1,u_1}, \ldots, S_{k_m,u_m}$ produced by the Subset Difference method. These subsets are encoded in such a way that unrevoked users can anonymously prove their membership of one of them. Our technique allows doing this using a proof of *constant* size.

The intuition is as follows. In the generation of RL_t, for each $i \in \{1, \ldots, m\}$, if x_{k_i} (resp. x_{u_i}) denotes the primary (resp. secondary) root of S_{k_i,u_i}, the GM encodes S_{k_i,u_i} as a vector of group elements R_i that determines the levels of nodes x_{k_i} and x_{u_i} in the tree (which are called ϕ_i and ψ_i hereafter) and the identifiers $\mathcal{ID}(x_{k_i})$ and $\mathcal{ID}(x_{u_i})$. Then, the resulting vector R_i is authenticated by means of a structure-preserving signature Θ_i, which is included in RL_t and will be used in a set membership proof.

During the join protocol, users obtain from the GM a structure-preserving signature on a compact encoding C_v – which is computed as a concise commitment to a vector of node identifiers $(I_1, \ldots, I_\ell)$ – of the path $(I_1, \ldots, I_\ell)$ between their leaf v and the root ϵ. This path is encoded as a single group element.

The group manager uses two key pairs for the AHO structure-preserving signature. The first one is used during the join protocol to bind a group element X chosen by the user, who knows $x = \log_g(X)$, to the path from the user's leaf v to the root ϵ.

In order to anonymously prove his/her non-revocation, a group member $\mathcal{U}_i$ uses RL_t to determine the generic subset S_{k_l,u_l}, with $l \in \{1, \ldots, m\}$, where his/her leaf v_i lies. He/she commits to the corresponding vector of group elements R_l that encodes the node identifiers $\mathcal{ID}(x_{k_l})$ and $\mathcal{ID}(x_{u_l})$ of the primary and secondary roots of S_{k_l,u_l} at levels ϕ_l and ψ_l, respectively. If $(I_1, \ldots, I_\ell)$ identifies the path from his/her leaf v_i to ϵ, the unrevoked member $\mathcal{U}_i$ generates a membership proof for the subset S_{k_l,u_l} by proving that $\mathcal{ID}(x_{k_l}) = I_{\phi_l}$ and $\mathcal{ID}(x_{u_l}) \neq I_{\psi_l}$ (in other words, that x_{k_l} is an ancestor of v_i and x_{u_l} is not). To succinctly prove these statements, $\mathcal{U}_i$ uses the properties of the LY concise vector commitment scheme[8]. Finally, in order to convince the verifier that he used a legal element of RL_t, $\mathcal{U}_i$ follows the technique of [61] and proves knowledge of a signature Θ_l on the committed vector of group elements R_l. By doing so, $\mathcal{U}_i$ thus provides evidence that his/her leaf v_i is a member of some authorized subset S_{k_l,u_l} without revealing $l \in \{1, \ldots, m\}$.

In order to obtain the strongest flavor of anonymity (*i.e.*, where the adversary has access to a signature opening oracle), the scheme uses Kiltz's tag-based encryption scheme as in Groth's construction [134] exactly as we did in the previous construction. In non-frameability concerns, the group member $\mathcal{U}_i$ also generates a weak Boneh-Boyen signature (which yields a fully secure signature when combined with a one-time signature) using $x = \log_g(X)$, where $X \in \mathbb{G}$ is a group element certified by the GM and bound to the path $(I_1, \ldots, I_\ell)$ during the join protocol.

Description

As in standard security models for group signatures, we assume that, before joining the group, user $\mathcal{U}_i$ chooses a long term key pair $(\mathsf{usk}[i], \mathsf{upk}[i])$ and registers it in some PKI.

Setup(λ, N)**:** given a security parameter $\lambda \in \mathbb{N}$ and the number of users $N = 2^{\ell-1}$,

1. Choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$, with $g \leftarrow \mathbb{G}$.
2. Define $n_0 = 2$ and $n_1 = 5$. Generates key pairs $(sk_{\mathsf{AHO}}^{(0)}, pk_{\mathsf{AHO}}^{(0)})$ and $(sk_{\mathsf{AHO}}^{(1)}, pk_{\mathsf{AHO}}^{(1)})$ for the AHO signature in order to sign messages of n_0 and n_1 group elements,

[8] Note that no randomness is needed here since we do not rely on the hiding property of the commitment.

respectively. These key pairs are

$$pk_{\mathsf{AHO}}^{(d)} = \Big(G_r^{(d)}, H_r^{(d)}, G_z^{(d)} = G_r^{\gamma_z^{(d)}}, H_z^{(d)} = H_r^{\delta_z^{(d)}}, \\ \{G_i^{(d)} = G_r^{\gamma_i^{(d)}}, H_i^{(d)} = H_r^{\delta_i^{(d)}}\}_{i=1}^{n_d}, A^{(d)}, B^{(d)}\Big)$$

and $sk_{\mathsf{AHO}}^{(d)} = \big(\alpha_a^{(d)}, \alpha_b^{(d)}, \gamma_z^{(d)}, \delta_z^{(d)}, \{\gamma_i^{(d)}, \delta_i^{(d)}\}_{i=1}^{n_d}\big)$, where $d \in \{0,1\}$. These will be used to sign messages consisting of 2 and 5 group elements, respectively.

3. Generate a public key $ck = (g_1, \ldots, g_\ell, g_{\ell+2}, \ldots, g_{2\ell}) \in \mathbb{G}^{2\ell-1}$ for ℓ-dimension vectors of the LY concise vector commitment scheme. The trapdoor $g_{\ell+1}$ is not needed and can be discarded.
4. As a Groth-Sahai CRS for the NIWI proof system, select three vectors $\mathbf{f} = (\mathbf{f_1}, \mathbf{f_2}, \mathbf{f_3})$ such that $\mathbf{f_1} = (f_1, 1, g) \in \mathbb{G}^3$, $\mathbf{f_2} = (1, f_2, g) \in \mathbb{G}^3$, and $\mathbf{f_3} = \mathbf{f_1}^{\xi_1} \cdot \mathbf{f_2}^{\xi_2}$, where $f_1 = g^{\beta_1}, f_2 = g^{\beta_2}$ in $\mathbb{G}$ and random $\beta_1, \beta_2, \xi_1, \xi_2 \leftarrow \mathbb{Z}_p^*$. We also define the vector $\boldsymbol{\varphi} = \mathbf{f_3} \cdot (1, 1, g)$.
5. Choose random $(U, V) \leftarrow \mathbb{G}^2$ that, together with generators $f_1, f_2, g \in \mathbb{G}$, will form a public encryption key.
6. Select a strongly unforgeable one-time signature $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$.
7. Sets $\mathcal{S}_{\mathsf{GM}} := \big(sk_{\mathsf{AHO}}^{(0)}, sk_{\mathsf{AHO}}^{(1)}\big)$, $\mathcal{S}_{\mathsf{OA}} := \big(\beta_1, \beta_2\big)$ as authorities' private keys and the group public key is

$$\mathcal{Y} := \Big(g,\ pk_{\mathsf{AHO}}^{(0)},\ pk_{\mathsf{AHO}}^{(1)},\ ck,\ \mathbf{f},\ \boldsymbol{\varphi},\ (U, V),\ \Sigma\Big).$$

Join$^{(\mathsf{GM}, \mathcal{U}_i)}$**:** the GM and the prospective user $\mathcal{U}_i$ run the following protocol:

1. $\mathcal{U}_i$ draws $x \leftarrow \mathbb{Z}_p$ at random and computes $X = g^x$ which is sent to the GM. If $X \in \mathbb{G}$ already appears in some entry $\mathsf{transcript}_j$ of the database St_{trans}, J_{GM} halts and returns $\perp$ to $\mathcal{U}_i$.
2. The GM assigns to the user $\mathcal{U}_i$ an available leaf v of identifier $\mathcal{ID}(v)$ in the tree T. Let $x_1, \ldots, x_\ell$ be the path from the chosen leaf $x_\ell = v$ to the root $x_1 = \epsilon$ of T. Let also $(I_1, \ldots, I_\ell) = (\mathcal{ID}(x_1), \ldots, \mathcal{ID}(x_\ell))$ be the corresponding vector of identifiers (with $I_1 = 1$ and $I_\ell = \mathcal{ID}(v) \in \{N, \ldots, 2N-1\}$). Then, the GM does the following.
   (a) Compute a compact encoding $C_v = \prod_{\kappa=1}^{\ell} g_{\ell+1-\kappa}^{I_\kappa} = g_\ell^{I_1} \cdots g_1^{I_\ell}$ of $(I_1, \ldots, I_\ell)$.
   (b) Using $sk_{\mathsf{AHO}}^{(0)}$, generate an AHO signature $\sigma_v = (\theta_{v,1}, \ldots, \theta_{v,7})$ on the pair $(X, C_v) \in \mathbb{G}^2$ so as to bind C_v to the value X that identifies $\mathcal{U}_i$.
3. The GM sends $\mathcal{ID}(v) \in \{N, \ldots, 2N-1\}$ and C_v to $\mathcal{U}_i$ that halts if $\mathcal{ID}(v) \notin \{N, \ldots, 2N-1\}$ or if C_v is found incorrect. Otherwise, $\mathcal{U}_i$ sends an ordinary digital signature $sig_i = \mathrm{Sign}_{\mathsf{usk}[i]}\big(X||(I_1, \ldots, I_\ell)\big)$ to the GM.
4. The GM checks that $\mathrm{Verify}_{\mathsf{upk}[i]}\big((X||(I_1, \ldots, I_\ell)), sig_i\big) = 1$. If not, the GM aborts. Otherwise, it returns the structure-preserving signature σ_v to the user $\mathcal{U}_i$ and stores $\mathsf{transcript}_i = (X, \mathcal{ID}(v), C_v, \sigma_v, sig_i)$ in the database St_{trans}.

5. The user $\mathcal{U}_i$ defines his membership certificate cert_i as

$$\mathsf{cert}_i = \big(\mathcal{ID}(v), X, C_v, \sigma_v\big) \in \{N, \ldots, 2N-1\} \times \mathbb{G}^9,$$

where X will serve as the tag identifying $\mathcal{U}_i$. The membership secret sec_i is defined as $\mathsf{sec}_i = x \in \mathbb{Z}_p$.

Revoke$(\mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, t, \mathcal{R}_t)$**:** Parse $\mathcal{S}_{\mathsf{GM}}$ as $\S_{\mathsf{GM}} := \big(sk^{(0)}_{\mathsf{AHO}}, sk^{(1)}_{\mathsf{AHO}}\big)$ and do the following.

1. Using the covering algorithm of the SD method, find a cover of the unrevoked user set $\{1, \ldots, N\} \backslash \mathcal{R}_t$ as the union of disjoint subsets of the form $S_{k_1,u_1}, \ldots, S_{k_m,u_m}$, with $m \leq 2 \cdot |\mathcal{R}_t| - 1$.
2. For $i = 1$ to m, do the following.
   (a) Consider S_{k_i,u_i} as the difference between sub-trees rooted at an internal node x_{k_i} and one of its descendants x_{u_i}. Lets $\phi_i, \psi_i \in \{1, \ldots, \ell\}$ be the depths of x_{k_i} and x_{u_i}, respectively, in T assuming that the root ϵ is at depth 1. Encode S_{k_i,u_i} as a vector $\big(g_{\phi_i}, g_1^{\mathcal{ID}(x_{k_i})}, g_{\psi_i}, g^{\mathcal{ID}(x_{u_i})}\big)$.
   (b) In order to authenticate S_{k_i,u_i} and bind it to the revocation period t, use $sk^{(1)}_{\mathsf{AHO}}$ to generate a structure-preserving signature $\Theta_i = (\Theta_{i,1}, \ldots, \Theta_{i,7}) \in \mathbb{G}^7$ on the message $R_i = \Big(g^t, g_{\phi_i}, g_1^{\mathcal{ID}(x_{k_i})}, g_{\psi_i}, g^{\mathcal{ID}(x_{u_i})}\Big) \in \mathbb{G}^5$, where the period number t is interpreted as an element of $\mathbb{Z}_p$.

Returns the revocation data

$$RL_t = \Big(t,\ \mathcal{R}_t,\ \big\{\phi_i,\ \psi_i,\ \mathcal{ID}(x_{k_i}),\ \mathcal{ID}(x_{u_i}),\ \Theta_i = (\Theta_{i,1}, \ldots, \Theta_{i,7})\big\}_{i=1}^{m}\Big). \tag{2.9}$$

Sign$(\mathcal{Y}, t, RL_t, \mathsf{cert}_i, \mathsf{sec}_i, M)$**:** returns $\perp$ if $i \in \mathcal{R}_t$. Otherwise, to sign $M \in \{0,1\}^*$, generates a one-time signature key pair $(\mathsf{sk}, \mathsf{VK}) \leftarrow \mathcal{G}(\lambda)$. Parse the membership certificate cert_i as $\mathsf{cert}_i = \big(\mathcal{ID}(v_i), X, C_{v_i}, \sigma_{v_i}\big) \in \{N, \ldots, 2N-1\} \times \mathbb{G}^9$ and sec_i as $x \in \mathbb{Z}_p$. Let $\epsilon = x_1, \ldots, x_\ell = v_i$ denote the path connecting v_i to the root ϵ of T and let $(I_1, \ldots, I_\ell) = (\mathcal{ID}(x_1), \ldots, \mathcal{ID}(x_\ell))$ be the vector of node identifiers. First, $\mathcal{U}_i$ generates a commitment $com_{C_{v_i}}$ to the encoding C_{v_i} of the path $(I_1, \ldots, I_\ell)$ from v_i to the root. Then, he does the following.

1. Using RL_t, find the set S_{k_l,u_l}, with $l \in \{1, \ldots, m\}$, that contains the leaf v_i identified by $\mathcal{ID}(v_i)$. Let x_{k_l} and x_{u_l} denote the primary and secondary roots of S_{k_l,u_l} at depths ϕ_l and ψ_l, respectively. Since x_{k_l} is an ancestor of v_i but x_{u_l} is not, it must be the case that $I_{\phi_l} = \mathcal{ID}(x_{k_l})$ and $I_{\psi_l} \neq \mathcal{ID}(x_{u_l})$.
2. In order to prove that v_i belongs to S_{k_l,u_l} without leaking l, re-randomize the l-th AHO signature Θ_l contained in RL_t as $\{\Theta'_{l,i}\}_{i=1}^{7} \leftarrow \mathsf{ReRand}(pk^{(1)}_{\mathsf{AHO}}, \Theta_l)$. Then, commit to the l-th revocation message

$$R_l = (R_{l,1}, \ldots, R_{l,5}) = \big(g^t, g_{\phi_l}, g_1^{\mathcal{ID}(x_{k_l})}, g_{\psi_l}, g^{\mathcal{ID}(x_{u_l})}\big) \tag{2.10}$$

and its signature $\Theta'_l = (\Theta'_{l,1}, \ldots, \Theta'_{l,7})$ by computing Groth-Sahai commitments $\{com_{R_{l,\tau}}\}_{\tau=2}^{5}$ and $\{com_{\Theta'_{l,j}}\}_{j\in\{1,2,5\}}$ to $\{R_{l,\tau}\}_{\tau=2}^{5}$ and $\{\Theta'_{l,j}\}_{j\in\{1,2,5\}}$ respectively.

(a) To prove that $I_{\phi_l} = \mathcal{ID}(x_{k_l})$, compute $W_{\phi_l} = \prod_{\kappa=1,\ \kappa \neq \phi_l}^{\ell} g_{\ell+1-\kappa+\phi_l}^{I_\kappa}$ that satisfies the equality $e(g_{\phi_l}, C_{v_i}) = e(g_1, g_\ell)^{I_{\phi_l}} \cdot e(g, W_{\phi_l})$. Then, generate a Groth-Sahai commitment $com_{W_{\phi_l}}$ to W_{ϕ_l}. Compute a NIWI proof that committed variables $(R_{l,2}, R_{l,3}, C_{v_i}, W_{\phi_l})$ satisfy

$$e(R_{l,2}, C_{v_i}) \quad = \quad e(R_{l,3}, g_\ell) \cdot e(g, W_{\phi_l}). \tag{2.11}$$

We denote by $\pi_{eq} \in \mathbb{G}^9$ the proof for the quadratic equation (2.11).

(b) To prove that $I_{\psi_l} \neq \mathcal{ID}(x_{u_l})$, compute $W_{\psi_l} = \prod_{\kappa=1,\ \kappa \neq \psi_l}^{\ell} g_{\ell+1-\kappa+\psi_l}^{I_\kappa}$ that satisfies $e(g_{\psi_l}, C_{v_i}) = e(g_1, g_\ell)^{I_{\psi_l}} \cdot e(g, W_{\psi_l})$. Then, compute a Groth-Sahai commitment $com_{W_{\psi_l}}$ to W_{ψ_l} as well as commitments com_{Γ_l} and $\{com_{\Psi_{l,\tau}}\}_{\tau \in \{0,1,2\ell\}}$ to the group elements

$$(\Gamma_l, \Psi_{l,0}, \Psi_{l,1}, \Psi_{l,2\ell}) = \left(g^{1/(I_{\psi_l} - \mathcal{ID}(x_{u_l}))}, g^{I_{\psi_l}}, g_1^{I_{\psi_l}}, g_{2\ell}^{I_{\psi_l}}\right).$$

The next step is to generate a NIWI proof that the committed group elements $(R_{l,4}, R_{l,5}, C_{v_i}, \Gamma_l, \Psi_{l,0}, \Psi_{l,1}, \Psi_{l,2\ell})$ satisfy

$$e(R_{l,4}, C_{v_i}) = e(\Psi_{l,1}, g_\ell) \cdot e(g, W_{\psi_l}), \tag{2.12}$$

$$e(\Psi_{l,0}/R_{l,5}, \Gamma_l) = e(g, g), \tag{2.13}$$

$$e(\Psi_{l,1}, g) = e(g_1, \Psi_{l,0}), \tag{2.14}$$

$$e(\Psi_{l,2\ell}, g) = e(g_{2\ell}, \Psi_{l,0}). \tag{2.15}$$

We denote this NIWI proof by $\pi_{neq} = (\pi_{neq,1}, \pi_{neq,2}, \pi_{neq,3}, \pi_{neq,4})$. Since the first two equations (2.12) and (2.13) are quadratic, $\pi_{neq,1}$ and $\pi_{neq,2}$ consist of 9 elements each. The last two equations (2.14) and (2.15) are linear and both cost 3 elements to prove.

3. Provide evidence that the tuple R_l of (2.10) is a certified revocation message for period t: namely, compute a NIWI proof π_{R_l} that committed message elements $\{R_{l,\tau}\}_{\tau=2}^{5}$ and signature components $\{\Theta'_{l,j}\}_{j \in \{1,2,5\}}$ satisfy the equations

$$\begin{aligned} A^{(1)} \cdot e(\Theta'_{l,3}, \Theta'_{l,4})^{-1} \cdot e(G_1^{(1)}, g^t)^{-1} &= e(G_z^{(1)}, \Theta'_{l,1}) \cdot e(G_r^{(1)}, \Theta'_{l,2}) \cdot \prod_{\tau=2}^{5} e(G_\tau^{(1)}, R_{l,\tau}), \\ B^{(1)} \cdot e(\Theta'_{l,6}, \Theta'_{l,7})^{-1} \cdot e(H_1^{(1)}, g^t)^{-1} &= e(H_z^{(1)}, \Theta'_{l,1}) \cdot e(H_r^{(1)}, \Theta'_{l,5}) \cdot \prod_{\tau=2}^{5} e(H_\tau^{(1)}, R_{l,\tau}). \end{aligned} \tag{2.16}$$

Since $\{\Theta'_{l,j}\}_{j \in \{3,4,6,7\}}$ are constants, equations (2.16) are both linear and thus require 3 elements each. Hence, π_{R_l} takes 6 elements altogether.

4. Let $\sigma_{v_i} = (\theta_{v_i,1}, \ldots, \theta_{v_i,7})$ be the AHO signature on (X, C_{v_i}). Re-randomize σ_{v_i} as $\{\theta'_{v_i,j}\}_{j=1}^{7} \leftarrow \mathsf{ReRand}(pk_{\mathsf{AHO}}^{(0)}, \sigma_{v_i})$ and generate commitments $\{com_{\theta'_{v_i,j}}\}_{j \in \{1,2,5\}}$ to $\{\theta'_{v_i,j}\}_{j \in \{1,2,5\}}$ as well as a commitment com_X to X. Then, generate a NIWI proof $\pi_{\sigma_{v_i}}$ that committed variables satisfy the verification equations

$$A^{(0)} \cdot e(\theta'_{l,3}, \theta'_{l,4})^{-1} = e(G_z^{(0)}, \theta'_{l,1}) \cdot e(G_r^{(0)}, \theta'_{l,2}) \cdot e(G_1^{(0)}, X) \cdot e(G_2^{(0)}, C_{v_i}),$$

$$B^{(0)} \cdot e(\theta'_{l,6}, \theta'_{l,7})^{-1} = e(H_z^{(0)}, \theta_{l,1}) \cdot e(H_r^{(0)}, \theta'_{l,5}) \cdot e(H_1^{(0)}, X) \cdot e(H_2^{(0)}, C_{v_i}).$$

Since these equations are linear, $\pi_{\sigma_{v_i}}$ requires 6 group elements.

5. Using VK as a tag, compute a tag-based encryption [160] of X by drawing random exponents $z_1, z_2 \leftarrow \mathbb{Z}_p$ at random and setting

$$(\mathrm{Y}_1, \mathrm{Y}_2, \mathrm{Y}_3, \mathrm{Y}_4, \mathrm{Y}_5) = \big(f_1^{z_1}, f_2^{z_2}, X \cdot g^{z_1+z_2}, (g^{\mathsf{VK}} \cdot U)^{z_1}, (g^{\mathsf{VK}} \cdot V)^{z_2}\big).$$

6. Generate a NIZK proof that $com_X = (1,1,X) \cdot \mathbf{f_1}^{w_{X,1}} \cdot \mathbf{f_2}^{w_{X,2}} \cdot \mathbf{f_3}^{w_{X,3}}$ and $(\mathrm{Y}_1, \mathrm{Y}_2, \mathrm{Y}_3)$ are BBS encryptions of the same value X. If we write $\mathbf{f_3} = (f_{3,1}, f_{3,2}, f_{3,3})$, the commitment com_X can be written as $(f_1^{w_{X,1}} \cdot f_{3,1}^{w_{X,3}}, f_2^{w_{X,2}} \cdot f_{3,2}^{w_{X,3}}, X \cdot g^{w_{X,1}+w_{X,2}} \cdot f_{3,3}^{w_{X,3}})$, so that we have

$$com_X \cdot (\mathrm{Y}_1, \mathrm{Y}_2, \mathrm{Y}_3)^{-1} = \big(f_1^{\chi_1} \cdot f_{3,1}^{\chi_3},\ f_2^{\chi_2} \cdot f_{3,2}^{\chi_3},\ g^{\chi_1+\chi_2} \cdot f_{3,3}^{\chi_3}\big) \tag{2.17}$$

with $\chi_1 = w_{X,1} - z_1, \chi_2 = w_{X,2} - z_2, \chi_3 = w_{X,3}$. Compute commitments to $\{\chi_j\}_{j=1}^3$ as $com_{\chi_j} = \boldsymbol{\varphi}^{\chi_j} \cdot \mathbf{f_1}^{w_{\chi_j,1}} \cdot \mathbf{f_2}^{w_{\chi_j,2}}$, with $w_{\chi_j,1}, w_{\chi_j,2} \xleftarrow{\$} \mathbb{Z}_p$ for $j \in \{1,2,3\}$ and generate proofs $\{\pi_{eq\text{-}com,j}\}_{j=1}^3$ that χ_1, χ_2, χ_3 satisfy the three linear relations (2.17). These latter proofs $\{\pi_{eq\text{-}com,j}\}_{j=1}^3$ cost 2 elements each.

7. Compute a Boneh-Boyen signature $\sigma_{\mathsf{VK}} = g^{1/(x+\mathsf{VK})}$ on VK and a commitment $com_{\sigma_{\mathsf{VK}}}$ to σ_{VK}. Then, generate a NIWI proof $\pi_{\sigma_{\mathsf{VK}}} = (\boldsymbol{\pi}_{\sigma_{\mathsf{VK}},1}, \boldsymbol{\pi}_{\sigma_{\mathsf{VK}},2}, \boldsymbol{\pi}_{\sigma_{\mathsf{VK}},3}) \in \mathbb{G}^9$ that the committed variables $(\sigma_{\mathsf{VK}}, X) \in \mathbb{G}^2$ satisfy $e(\sigma_{\mathsf{VK}}, X \cdot g^{\mathsf{VK}}) = e(g,g)$.

8. Compute a one-time signature $\sigma_{ots} = \mathcal{S}(\mathsf{sk}, (M, RL_t, \mathrm{Y}_1, \mathrm{Y}_2, \mathrm{Y}_3, \mathrm{Y}_4, \mathrm{Y}_5, \Omega, \mathbf{com}, \mathbf{\Pi}))$ where $\Omega = \{\Theta'_{l,i}, \theta'_{l,i}\}_{i \in \{3,4,6,7\}}$ and

$$\begin{aligned}\mathbf{com} &= \big(com_{C_{v_i}}, com_X, \{com_{R_{l,\tau}}\}_{\tau=2}^5, com_{W_{\phi_l}}, com_{W_{\psi_l}}, com_{\Gamma_l}, \{com_{\Psi_{l,\tau}}\}_{\tau \in \{0,1,2\ell\}}, \\ &\qquad \{com_{\Theta'_{l,j}}\}_{j \in \{1,2,5\}}, \{com_{\theta'_{l,j}}\}_{j \in \{1,2,5\}}, \{com_{\chi_j}\}_{j=1}^3, com_{\sigma_{\mathsf{VK}}}\big), \\ \mathbf{\Pi} &= \big(\pi_{eq}, \pi_{neq}, \pi_{R_l}, \pi_{\sigma_{v_i}}, \{\pi_{eq\text{-}com,j}\}_{j=1}^3, \pi_{\sigma_{\mathsf{VK}}}\big).\end{aligned}$$

Return the signature

$$\sigma = \big(\mathsf{VK}, \mathrm{Y}_1, \mathrm{Y}_2, \mathrm{Y}_3, \mathrm{Y}_4, \mathrm{Y}_5, \Omega, \mathbf{com}, \mathbf{\Pi}, \sigma_{ots}\big). \tag{2.18}$$

Verify$(\sigma, M, t, RL_t, \mathcal{Y})$**:** parse σ as in (2.18). If $(\mathrm{Y}_1, \mathrm{Y}_2, \mathrm{Y}_3, \mathrm{Y}_4, \mathrm{Y}_5)$ is not a well-formed tag-based encryption (that is, if $e(\mathrm{Y}_1, g^{\mathsf{VK}} \cdot U) \neq e(f_1, \mathrm{Y}_4)$ or $e(\mathrm{Y}_2, g^{\mathsf{VK}} \cdot V) \neq e(f_2, \mathrm{Y}_5)$) or if $\mathcal{V}(\mathsf{VK}, (M, RL_t, \mathrm{Y}_1, \mathrm{Y}_2, \mathrm{Y}_3, \mathrm{Y}_4, \mathrm{Y}_5, \Omega, \mathbf{com}, \mathbf{\Pi}), \sigma_{ots}) = 0$, return 0. Then, return 1 if all proofs properly verify. Otherwise, return 0.

Open$(M, t, RL_t, \sigma, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, St)$**:** parse σ as above and return $\perp$ if $\mathsf{Verify}(\sigma, M, t, RL_t, \mathcal{Y}) = 0$. Otherwise, given $\mathcal{S}_{\mathsf{OA}} = (\beta_1, \beta_2)$, compute $\tilde{X} = \mathrm{Y}_3 \cdot \mathrm{Y}_1^{-1/\beta_1} \cdot \mathrm{Y}_2^{-1/\beta_2}$. In the database St_{trans}, find a record $\langle i, \mathsf{transcript}_i = (X_i, \mathcal{ID}(v_i), C_{v_i}, \sigma_{v_i}, sig_i)\rangle$ such that $X_i = \tilde{X}$. If no such record exists in St_{trans}, returns $\perp$. Otherwise, return i.

At first glance, the variable $\Psi_{l,2\ell}$ and the proof of the second equality (2.14) may seem unnecessary in step 2.b of the signing algorithm. However, this element plays a crucial role when it comes to proving the security under the ℓ-FlexDHE assumption. Indeed, the proof of security against misidentification attacks ceases to go through if we remove $\Psi_{l,2\ell}$ and its corresponding proof.

Efficiency

As far as efficiency goes, each entry of RL_t contains 7 group elements and two node identifiers of $O(\log N)$ bits each. If $\lambda_{\mathbb{G}}$ is the bitlength of a group element, we have $\log N \ll \lambda_{\mathbb{G}}/2$ (since $\lambda \leq \lambda_{\mathbb{G}}$ and N is polynomial), so that the number of bits of RL_t is bounded by $2 \cdot |\mathcal{R}_t| \cdot (7 \cdot \lambda_{\mathbb{G}} + 2 \log N + 2 \log \log N) < 2 \cdot |\mathcal{R}_t| \cdot (9\lambda_{\mathbb{G}})$ bits. The size of RL_t is thus bounded by that of $18 \cdot |\mathcal{R}_t|$ group elements.

Unlike our first scalable construction [180], group members only need to store 9 group elements in their membership certificate. As far as the size of signature goes, **com** and $\mathbf{\Pi}$ require 66 and 60 group elements, respectively. If the one-time signature of [133] is used, VK and σ_{ots} consist of 3 elements of $\mathbb{G}$ and 2 elements of $\mathbb{Z}_p$, respectively. The global size σ amounts to that of 144 group elements, which is about 50% longer than [180]. In comparison with [134] (which does not natively support revocation), signatures are only longer by a factor of 3. At the 128-bit security level, each group element should have a 512-bit representation and a signature takes 9 kB.

Verifying signatures takes time $O(1)$. The signer has to compute $2\ell = O(\log N)$ exponentiations to obtain W_{ϕ_l} and W_{ψ_l} at the beginning of each period. Note that these exponentiations involve short exponents of $O(\log N)$ bits each. Hence, computing W_{ϕ_l} and W_{ψ_l} requires $O(\log^2 N)$ multiplications in $\mathbb{G}$. For this reason, since $\log^2 N \ll \lambda$ (as long as $N \ll 2^{\lambda^{1/2}}$), this cost is dominated by that of a single exponentiation in $\mathbb{G}$.

Security

The security of the scheme relies on the same assumptions as in our first revocable group signature [180] (namely, the q-SFP, q-SDH and DLIN assumptions) and the ℓ-FlexDHE assumption. While we need an addition non-standard assumption, we only need the ℓ-FlexDHE assumption to hold for small values of the parameter $\ell = \log N$, where N is the maximal number of users.

In the article [179, Appendix C], we suggest a variant of the scheme where the ℓ-FlexDHE assumption is replaced by an assumption of constant size, introduced by Laguillaumie *et al.* [167], at the expense of increasing the group public key size from $O(\log N)$ to $O(\log^2 N)$. This is achieved by replacing the concise vector commitment of Libert and Yung [188] by the one of Catalano and Fiore [75], which relies on the CDH assumption instead of the ℓ-DHE assumption but has a longer commitment key. By applying the results of Abe *et al.* [2] to our modified scheme [179, Appendix C], it is further possible to construct a revocable group signature with $O(\log^2 N)$-size group public keys which only relies on simple assumptions in the standard model.

In a follow-up work, Attrapadung *et al.* [20] used a different mechanism from the broadcast encryption literature – due to Attrapadung, Libert and de Panafieu [24, 21] – to achieve an efficiency tradeoff which is exactly dual to ours. While we obtain membership certificates and revocation lists made of $O(1)$ and $O(r)$ group elements, respectively, Attrapadung *et al.* [20] perform the other way around with $O(1)$-size revocation lists and $O(R)$-size membership certificates, where R is an upper bound on the number of revoked users. However, the maximal number R of revoked users must be fixed in advance even if it is much smaller than the total number of users N. Similar results were obtained by Nakanishi and Funabiki [204].

2.3 Conclusion

This chapter presented two important applications of structure-preserving cryptographic primitives in the design of anonymity-related cryptographic mechanisms. One of our contributions was the first reasonably efficient construction [81] – which was proposed at the same time as (and independently of) Fuchsbauer's automorphic signatures [112] – of the primitive, initially introduced by Groth [133], that was subsequently named "structure-preserving signature" by Abe *et al.* [6, 4]. This construction allowed us to obtain the first fully non-interactive group encryption system in the standard model and also immediately implied the first group signatures with concurrent join in the standard model [157]. Together with other techniques (such as the NNL framework [205] and our construction of concise vector commitments [188]), the optimized SPS scheme of Abe *et al.* [6] also enabled the design of a new revocation mechanism for group signature schemes in the standard model.

Structure-preserving signatures were also used in other results of mine [176, 173] on privacy-preserving primitives which are not discussed in this manuscript. In collaboration with Marc Joye, Moti Yung and Thomas Peters, we built on the Abe *et al.* [6] system to construct a group encryption scheme [176] with refined tracing capabilities similar to those of traceable signatures [155]: specifically, the opening authority can disclose a user-specific trapdoor that makes it possible to trace all ciphertexts encrypted for a given suspicious user without affecting the privacy of well-behaved users. Together with Marc Joye, we also designed a partially structure-preserving identity-based encryption (IBE) scheme [173] – where "partially" means that identities are still encoded as bitstrings (rather than group elements) but encrypted messages live in the source group $\mathbb{G}$ of the bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ instead of the target group $\mathbb{G}_T$ as in most IBE schemes in the standard model [248] – and used it to construct the first efficient standard model realization of group signatures with message-dependent opening (GS-MDO) [233]. In short GS-MDO schemes, as introduced by Sakai *et al.* [233], are group signatures where the opening authority can only open signatures for which a separate authority has released a message specific trapdoor. Sakai *et al.* [233] showed that GS-MDO implies identity-based encryption, which raised the intuition that realizing GS-MDO schemes in the standard model requires a structure-preserving IBE. In [173], we showed that a partially structure-preserving IBE suffices for this purpose and we built a GS-MDO scheme with logarithmic signature size in the standard model.

Chapter 3

Constructions of Non-Malleable Primitives from Structure-Preserving Cryptography

In the last three years, a large body of work has analyzed the feasibility and the efficiency of structure-preserving signatures (SPS) [133, 81, 112, 6, 4, 5, 63, 82, 142, 2, 3], public-key encryption [65] and commitments schemes [135, 7].

In this chapter, we consider applications of structure-preserving signatures in the design of non-malleable protocols such as non-interactive non-malleable commitments or chosen-ciphertext-secure public-key encryption. Paradoxically, this is achieved by first considering structure-preserving signatures which are intentionally made malleable. We consider SPS schemes with linearly homomorphic properties and argue that such primitives have many applications, even independently of Groth-Sahai proofs.

3.0.1 Linearly Homomorphic Structure-Preserving Signatures

The concept of homomorphic signatures can be traced back to Desmedt [97] while proper definitions remained lacking until the work of Johnson *et al.* [148]. Since then, constructions have appeared for various kinds of homomorphisms (see [11] and references therein).

Linearly Homomorphic Schemes. Linearly homomorphic signatures are an important class of homomorphic signatures for arithmetic functions, whose study was initiated by Boneh, Freeman, Katz and Waters [48]. While initially motivated by applications to network coding [48], they are also useful in proofs of storage [13, 16] or in verifiable computation mechanisms, when it comes to authenticate servers' computations on outsourced data (see, *e. g.*, [11]). The recent years, much attention was given to the notion and a variety of constructions [120, 23, 47, 46, 77, 78, 111, 25, 26] based on various assumptions have been studied.

Structure-Preserving Signatures Made Homomorphic. In collaboration with Thomas Peters, Marc Joye and Moti Yung [177], we put forth the notion of linearly homomorphic structure-preserving signatures (LHSPS). While structure-preserving signatures and linearly homomorphic signatures have both been studied before, simultaneously combining the homomorphic and structure-preserving properties turns out to be useful and non-trivial. As we will see in this chapter, such a combination has unexpected applications that are not known to be possible with only one of these two properties individually. In particular, we describe applications of LHSPS schemes *beyond* their compatibility with the Groth-Sahai techniques. These signature schemes function exactly like ordinary homomorphic signatures with the additional restriction that signatures and messages only consist of (vectors of) group elements whose discrete logarithms may not be available. We describe three constructions and prove their security under well-established assumptions in bilinear groups.

Our first scheme's starting point is the one-time (regular) SPS scheme of Abe *et al.* [6]. By removing certain public key components, we obtain the desired linear homomorphism, and prove the security using information-theoretic arguments as in [6]. The key observation here is that, as long as the adversary does not output a signature on a linear combination of previously signed vectors, it will be unable to sign its target vector in the same way as the reduction would, because certain private key components will remain perfectly hidden.

Our initial scheme inherits the one-time restriction of the scheme in [6] in that only one linear subspace can be safely signed with a given public key. Nevertheless, we can extend it to build a full linearly homomorphic SPS system. To this end, we suitably combine our first scheme with Waters signatures [248]. Here, Waters signatures are used as a resting ground for fresh random exponents which are introduced in each signed vector and help us refresh the state of the system and apply each time the same argument as in the one-time scheme. We also present techniques to turn the scheme into a fully randomizable one, where a derived signature has the same distribution as a directly signed message.

3.0.2 Applications

Verifiable computation on encrypted data. First, we show that the primitive enables verifiable computation mechanisms on encrypted data.[1] Specifically, it allows a client to store encrypted files on an untrusted remote server. While the dataset is encrypted using an additively homomorphic encryption scheme, the server is able to blindly compute linear functions on the original data and provide the client with a short homomorphically derived signature vouching for the correctness of the computation. This is achieved by having the client sign each ciphertext using a homomorphic SPS scheme and handing the resulting signatures to the server at the beginning. After this initial phase, the client only needs to store a short piece of information, no matter how large the file is. Still, he remains able to authenticate linear functions on his data and the whole process is completely non-interactive.

Non-malleable commitments to group elements. As a more surprising application, we show that LHSPS schemes generically yield non-malleable [102] trapdoor commitments to group elements. We actually construct a simulation-sound trapdoor commitment [116] — a primitive known (by [116, 195]) to imply re-usable non-malleable commitments with respect to opening [94] — from any linearly homomorphic SPS satisfying a relatively mild condition. To our knowledge, we thus obtain the first constant-size trapdoor commitments to group elements providing re-usable non-malleability with respect to opening. Previous non-interactive commitments to group elements were either malleable [138, 135] or inherently length-increasing [108]: if we disregard the trivial solution consisting of hashing the message first (which is not an option when we want to allow for efficient proofs of knowl-

[1] Our goals are very different from those of [119], where verifiable computation on homomorphically encrypted data is also considered. We do not seek to outsource computation but rather save the client from storing large datasets.

edge of an opening), no general technique has been known, to date, for committing to many group elements at once using a short commitment string.

In the structure-preserving case, our transformation is purely generic as it applies to a template which any linearly homomorphic SPS necessarily satisfies in symmetric bilinear groups. We also generalize the construction so as to build simulation-sound trapdoor commitments to vectors from any pairing-based (non-structure-preserving) linearly homomorphic signature. In this case, the conversion is only semi-generic as it imposes conditions which are only met by pairing-based systems for the time being: essentially, we need the underlying signature scheme to operate over groups of finite, public order. While only partially generic, this construction of non-malleable commitments from linearly homomorphic signatures is somewhat unexpected considering that the terms "non-malleability" and "homomorphism" are antagonistic, and thus may be considered incompatible.

Constant-Size Quasi-Adaptive NIZK Proofs for Linear Subspaces. Our LHSPS schemes also allowed us [178] to construct constant-size QA-NIZK arguments of linear subspace membership. Given a $t \times n$ matrix of group elements of rank $t < n$, the QA-NIZK proofs of Jutla and Roy [151] save $\Omega(t)$ group elements compared to Groth-Sahai. In [178], we gave QA-NIZK arguments for proving the same statement using a *constant* number group elements, regardless of the number of equations or the number of variables. Our one-time LHSPS system immediately gives QA-NIZK arguments of linear subspace membership comprised of only 3 group elements under the DLIN assumption (and 2 group elements under the SXDH assumption). While our constant-size QA-NIZK arguments are malleable in their simplest version, they readily extend – at minimal cost – to provide a form of one-time simulation-soundness defined by Jutla and Roy [150]. Moreover, we describe a construction of *unbounded* simulation-sound QA-NIZK argument based on our randomizable LHSPS system. Unlike previous unbounded simulation-sound Groth-Sahai-based proofs, our construction does not involve quadratic pairing product equations and does not rely on a chosen-ciphertext-secure encryption scheme.

Our constant-size QA-NIZK argument systems allowed us [178] to design new and improved CCA2-secure encryption schemes. In particular, we could significantly optimize the adaptively secure non-interactive threshold versions of the Cramer-Shoup cryptosystem given by Libert and Yung [191]. We also built an efficient CCA2-secure keyed-homomorphic encryption scheme. Keyed-homomorphic encryption is a primitive, suggested by Emura *et al.*[104], which allows reconciling homomorphism and IND-CCA2 security. The idea of Emura *et al.*[104] is that homomorphic operations can only be carried out using a dedicated evaluation key. A keyed homomorphic scheme should be designed so as to be chosen-ciphertext-secure against any adversary that is withheld access to the evaluation key. At the same time, the evaluation key does not enable decryption and IND-CCA1 security should be preserved even if this evaluation key is made available to the adversary. The keyed homomorphic constructions of Emura *et al.*[104] are only known to satisfy a relaxed definition of security where the adversary is only given access to a restricted homomorphic evaluation oracle. Using our unbounded simulation-sound QA-NIZK proofs, we were able [178] to build a keyed homomorphic encryption scheme satisfying the strongest definition of chosen-ciphertext security given in [104]. At the same time, our construction enables threshold decryption, as shown in [178], which is a useful capability in many applications of homomorphic encryption. Our results were recently improved by Jutla and Roy [153, 152] who gave even shorter QA-NIZK proofs [153] of linear subspace membership and improved unbounded simulation-sound constructions [152].

3.1 Linearly Homomorphic Structure-Preserving Signatures

3.1.1 Definitions for Linearly Homomorphic Signatures

Let $(\mathbb{G}, \mathbb{G}_T)$ be a configuration of (multiplicatively written) groups of prime order p over which a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is efficiently computable.

We consider linearly homomorphic signatures for which the message space $\mathcal{M}$ consists of pairs $\mathcal{M} := \mathcal{T} \times \mathbb{G}^n$, for some $n \in \mathbb{N}$, where $\mathcal{T}$ is a tag space. We remark that, in the applications considered in this paper, tags do not need to be group elements. We thus allow them to be arbitrary strings.

Definition 5. *A linearly homomorphic structure-preserving signature (LHSPS) over* $(\mathbb{G}, \mathbb{G}_T)$ *is a tuple of efficient algorithms* $\Sigma = (\mathsf{Keygen}, \mathsf{Sign}, \mathsf{SignDerive}, \mathsf{Verify})$ *for which the message space is* $\mathcal{M} := \mathcal{T} \times \mathbb{G}^n$*, for some* $n \in \mathsf{poly}(\lambda)$ *and some set* $\mathcal{T}$*, and such that:*

$\mathsf{Keygen}(\lambda, n)$: *is a randomized algorithm that takes in a security parameter* $\lambda \in \mathbb{N}$ *and an integer* $n \in \mathsf{poly}(\lambda)$ *denoting the dimension of vectors to be signed. It outputs a key pair* (pk, sk) *and the description of a tag (i.e., a file identifier) space* $\mathcal{T}$*.*

$\mathsf{Sign}(\mathsf{sk}, \tau, \mathbf{M})$: *is a possibly probabilistic algorithm that takes in a private key sk, a file identifier* $\tau \in \mathcal{T}$ *and a vector* $\mathbf{M} \in \mathbb{G}^n$*. It outputs a signature* $\sigma \in \mathbb{G}^{n_s}$*, for some* $n_s \in \mathsf{poly}(\lambda)$ *determined by* pk*.*

$\mathsf{SignDerive}(\mathsf{pk}, \tau, \{(\omega_i, \sigma^{(i)})\}_{i=1}^{\ell})$: *is a (possibly probabilistic) signature derivation algorithm. It takes as input a public key* pk*, a file identifier* τ *as well as* ℓ *pairs* $(\omega_i, \sigma^{(i)})$*, each of which consists of a weight* $\omega_i \in \mathbb{Z}_p$ *and a signature* $\sigma^{(i)} \in \mathbb{G}^{n_s}$*. The output is a signature* $\sigma \in \mathbb{G}^{n_s}$ *on the vector* $\mathbf{M} = \prod_{i=1}^{\ell} \mathbf{M}_i^{\omega_i}$*, where* $\sigma^{(i)}$ *is a signature on* $\mathbf{M}_i$*.*

$\mathsf{Verify}(\mathsf{pk}, \tau, \mathbf{M}, \sigma)$: *is a deterministic algorithm that takes in a public key* pk*, a file identifier* $\tau \in \mathcal{T}$*, a signature* σ *and a vector* $\mathbf{M}$*. It outputs* 1 *if* σ *is deemed valid and* 0 *otherwise.*

Correctness is expressed by imposing that, for all security parameters $\lambda \in \mathbb{N}$, all integers $n \in \mathsf{poly}(\lambda)$ and all triples $(\mathsf{pk}, \mathsf{sk}, \mathcal{T}) \leftarrow \mathsf{Keygen}(\lambda, n)$, the following holds:

1. For all identifiers $\tau \in \mathcal{T}$ and all n-vectors $\mathbf{M} \in \mathbb{G}^n$, if $\sigma = \mathsf{Sign}(\mathsf{sk}, \tau, \mathbf{M})$, then we have $\mathsf{Verify}(\mathsf{pk}, \tau, \mathbf{M}, \sigma) = 1$.

2. For all identifiers $\tau \in \mathcal{T}$, any $\ell > 0$ and any set of triples $\{(\omega_i, \sigma^{(i)}, \mathbf{M}_i)\}_{i=1}^{\ell}$, if we have $\mathsf{Verify}(\mathsf{pk}, \tau, \mathbf{M}_i, \sigma^{(i)}) = 1$ for each $i \in \{1, \dots, \ell\}$, then

$$\mathsf{Verify}\big(\mathsf{pk}, \tau, \prod_{i=1}^{\ell} \mathbf{M}_i^{\omega_i}, \mathsf{SignDerive}(\mathsf{pk}, \tau, \{(\omega_i, \sigma^{(i)})\}_{i=1}^{\ell})\big) = 1.$$

In our constructions, n_s will be a constant which does not depend on the dimension n of signed vectors. This will play a crucial role in certain application like short quasi-adaptive NIZK proofs of linear subspace membership.

Security. At first, the very name of the primitive may sound almost self-contradictory when it comes to formally define its security. Indeed, the security of a linearly homomorphic scheme [48] notably requires that it be infeasible to publicly compute a signature on a vector outside the linear span of originally signed vectors. The problem is that, when vector entries live in a discrete-logarithm hard group, deciding whether several vectors are independent or not is believed to be a hard problem. Yet, this will not prevent us from applying new techniques and constructing schemes with security proofs under simple assumptions. In the security proof of our first construction, the reduction will be able to detect when the adversary has won using the private key of the system.

In linearly homomorphic signatures, we use the same definition of unforgeability as in [25]. This definition implies security in the stronger model used by Freeman [111] since the adversary can interleave signing queries for individual vectors belonging to distinct subspaces. Moreover, file identifiers can be chosen by the adversary (which strengthens the definition of [48]) and are not assumed to be random. As a result, a file identifier can be a low-entropy, easy-to-remember string such as the name of the dataset's owner.

Definition 6. *A linearly homomorphic SPS scheme $\Sigma = (\mathsf{Keygen}, \mathsf{Sign}, \mathsf{SignDerive}, \mathsf{Verify})$ is secure if no PPT adversary has non-negligible advantage in the game below:*

1. *The adversary $\mathcal{A}$ chooses an integer $n \in \mathbb{N}$ and sends it to the challenger who runs $\mathsf{Keygen}(\lambda, n)$ and obtains $(\mathsf{pk}, \mathsf{sk})$ before sending pk to $\mathcal{A}$.*
2. *On polynomially-many occasions, $\mathcal{A}$ can interleave the following kinds of queries.*
   - *Signing queries: $\mathcal{A}$ chooses a tag $\tau \in \mathcal{T}$ and a vector $\mathbf{M} \in \mathbb{G}^n$. The challenger picks a handle h and computes $\sigma \leftarrow \mathsf{Sign}(sk, \tau, \mathbf{M})$. It stores $(\mathsf{h}, (\tau, \mathbf{M}), \sigma)$ in a table T and returns h.*
   - *Derivation queries: $\mathcal{A}$ chooses a vector of handles $\mathsf{h} = (\mathsf{h}_1, \ldots, \mathsf{h}_k)$ and a set of coefficients $\{\omega_i\}_{i=1}^k$. The challenger retrieves the tuples $\{(\mathsf{h}_i, (\tau_i, \mathbf{M}_i), \sigma^{(i)})\}_{i=1}^k$ from T and returns $\perp$ if one of these does not exist or if there exists $i \in \{1, \ldots, k\}$ such that $\tau_i \neq \tau$. Otherwise, it computes the linear combination $\mathbf{M} = \prod_{i=1}^k \mathbf{M}_i^{\omega_i}$ and runs $\sigma' \leftarrow \mathsf{SignDerive}\big(\mathsf{pk}, \tau, \{(\omega_i, \sigma^{(i)})\}_{i=1}^k\big)$. It also chooses a handle h', stores $(\mathsf{h}', (\tau, \mathbf{M}), \sigma')$ in T and returns h' to $\mathcal{A}$.*
   - *Reveal queries: $\mathcal{A}$ chooses a handle h. If no tuple of the form $(\mathsf{h}, (\tau, \mathbf{M}), \sigma')$ exists in T, the challenger returns $\perp$. Otherwise, it returns σ' to $\mathcal{A}$ and adds $((\tau, \mathbf{M}), \sigma')$ to the set Q.*
3. *$\mathcal{A}$ outputs an identifier $\tau^\star$, a signature $\sigma^\star$ and a vector $\mathbf{M}^\star \in \mathbb{G}^n$. The adversary $\mathcal{A}$ wins if $\mathsf{Verify}(\mathsf{pk}, \tau^\star, \mathbf{M}^\star, \sigma^\star) = 1$ and one of the conditions below is satisfied:*
   - ○ *(Type I): $\tau^\star \neq \tau_i$ for any entry $(\tau_i, .)$ in Q and $\mathbf{M}^\star \neq (1_{\mathbb{G}}, \ldots, 1_{\mathbb{G}})$.*
   - ○ *(Type II): $\tau^\star = \tau_i$ for $k_i > 0$ entries $(\tau_i, .)$ in Q and $\mathbf{M}^\star \notin V_i$, where V_i denotes the subspace spanned by all vectors $\mathbf{M}_1, \ldots, \mathbf{M}_{k_i}$ for which an entry of the form $(\tau^\star, \mathbf{M}_j)$, with $j \in \{1, \ldots, k_i\}$, appears in Q.*

$\mathcal{A}$'s advantage is its probability of success taken over all coin tosses.

In our first scheme, we will consider a weaker notion of *one-time* security. In this notion, the adversary is limited to obtain signatures for only *one* linear subspace. In this case, there is no need for file identifiers and we assume that all vectors are assigned the identifier $\tau = \varepsilon$.

In the following, the adversary will be said *independent* if

- For any given tag τ, it is restricted to only query signatures on linearly independent vectors.
- Each vector is only queried at most once.

Non-independent adversaries are not subject to the above restrictions. It will be necessary to consider these adversaries in our construction of non-malleable commitments. Nevertheless, security against independent adversaries suffices for many applications —including encrypted cloud storage— since the signer can always append unit vectors to each newly signed vector.

At first, one may wonder how Definition 6 can be satisfied at all given that the challenger may not have an efficient way to check whether the adversary is successful. Indeed, in cryptographically useful discrete-logarithm-hard groups $\mathbb{G}$, deciding whether vectors $\{\mathbf{M}_i\}_i$ of $\mathbb{G}^n$ are linearly dependent is believed to be difficult when $n > 2$. However, it may be possible using some trapdoor information embedded in pk, especially if the adversary additionally outputs signatures on $\{\mathbf{M}_i\}_i$.

In some applications, it makes sense to consider a weaker attack model where a Type II adversary is only deemed successful if it outputs a convincing proof that its target vector $\mathbf{M}^\star$ is indeed independent of the vectors that were signed for the tag $\tau^\star$. The proof can be either a NIZK proof or, alternatively, a vector in the kernel of the matrix whose rows are the vectors that were signed for $\tau^\star$. We call such an adversary a *targeting* adversary.

3.2 Constructions of Linearly Homomorphic Structure-Preserving Signatures

As a warm-up, we begin by describing a one-time homomorphic signature, where a given public key allows signing only *one* linear subspace.

3.2.1 A One-Time Linearly Homomorphic Construction

The construction is based on a one-time structure-preserving signature described by Abe *et al.* [6, Appendix C.1] and the observation that this system can be made homomorphic by removing certain public key components.

In the description hereunder, since only one linear subspace can be signed for each public key, no file identifier τ is used. We thus set τ to be the empty string ε in all algorithms.

Keygen(λ, n): given a security parameter λ and the dimension $n \in \mathbb{N}$ of the subspace to be signed, choose bilinear group $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$. Then, choose generators $h, g_z, g_r, h_z \stackrel{\$}{\leftarrow} \mathbb{G}$. Pick $\chi_i, \gamma_i, \delta_i \stackrel{\$}{\leftarrow} \mathbb{Z}_p$, for $i = 1$ to n. Then, for each $i \in \{1, \ldots, n\}$, compute $g_i = g_z^{\chi_i} g_r^{\gamma_i}$, $h_i = h_z^{\chi_i} h^{\delta_i}$. The private key is $\mathsf{sk} = \{\chi_i, \gamma_i, \delta_i\}_{i=1}^n$ while the public key is defined to be

$$\mathsf{pk} = \big(g_z,\ h_r,\ h_z,\ h,\ \{g_i, h_i\}_{i=1}^n\big) \in \mathbb{G}^{2n+4}.$$

Sign$(\mathsf{sk}, \tau, (M_1, \ldots, M_n))$: to sign a vector $(M_1, \ldots, M_n) \in \mathbb{G}^n$ associated with the identifier $\tau = \varepsilon$ using $sk = \{\chi_i, \gamma_i, \delta_i\}_{i=1}^n$, compute the signature consists of $\sigma = (z, r, u) \in \mathbb{G}^3$, where

$$z = \prod_{i=1}^{n} M_i^{-\chi_i}, \qquad r = \prod_{i=1}^{n} M_i^{-\gamma_i}, \qquad u = \prod_{i=1}^{n} M_i^{-\delta_i}.$$

SignDerive$(\mathsf{pk}, \tau, \{(\omega_i, \sigma^{(i)})\}_{i=1}^{\ell})$: given the public key pk, a file identifier $\tau = \varepsilon$ and ℓ tuples $(\omega_i, \sigma^{(i)})$, parse each $\sigma^{(i)}$ as $\sigma^{(i)} = \big(z_i, r_i, u_i\big) \in \mathbb{G}^3$ for $i = 1$ to ℓ. Compute and return the derived signature $\sigma = (z, r, u) = \big(\prod_{i=1}^{\ell} z_i^{\omega_i}, \prod_{i=1}^{\ell} r_i^{\omega_i}, \prod_{i=1}^{\ell} u_i^{\omega_i}\big)$.

Verify$(\mathsf{pk}, \sigma, \tau, (M_1, \ldots, M_n))$: given a signature $\sigma = (z, r, u) \in \mathbb{G}^3$, a vector $(M_1, \ldots, M_n)$ and a file identifier $\tau = \varepsilon$, return 1 iff $(M_1, \ldots, M_n) \neq (1_{\mathbb{G}}, \ldots, 1_{\mathbb{G}})$ and (z, r, u) satisfy

$$1_{\mathbb{G}_T} = e(g_z, z) \cdot e(g_r, r) \cdot \prod_{i=1}^{n} e(g_i, M_i), \qquad 1_{\mathbb{G}_T} = e(h_z, z) \cdot e(h, u) \cdot \prod_{i=1}^{n} e(h_i, M_i). \tag{3.1}$$

The security proof relies on the fact that, while the signing algorithm is deterministic, signatures are not unique as each vector has an exponential number of valid signatures. However, the reduction can compute exactly one signature for each vector. At the same time, an adversary has no information about which specific signature the legitimate signer would compute on a vector outside the span of already signed vectors. Moreover, by obtaining two distinct signatures on a given vector, the reduction can readily solve a given instance of the SDP problem [81].

Theorem 3 ([177]). *The scheme is unforgeable if the SDP assumption holds in* $(\mathbb{G}, \mathbb{G}_T)$.

The scheme can be modified so as to work in asymmetric pairing configurations and the Double Pairing assumption.

One particularity of this scheme is that, even if the private key is available, it remains difficult to find two distinct signatures on the same vector if the SDP assumption holds: by dividing out the two signatures, one obtains the solution of an SDP instance (g_z, g_r, h_z, h_u) contained in the public key.

3.2.2 A Full-Fledged Linearly Homomorphic SPS Scheme

Our one-time construction can be upgraded to obtain a scheme allowing to sign an arbitrary number of linear subspaces. Here, each file identifier τ consists of a L-bit string. The construction builds on the observation that, in the scheme of Section 3.2.1, signatures (z, r, u) could be re-randomized by computing $(z \cdot g_r^{\theta}, r \cdot g_z^{-\theta}, u \cdot h_z^{-\log_h(g_r) \cdot \theta})$, with $\theta \xleftarrow{\$} \mathbb{Z}_p$, if $h_z^{-\log_h(g_r)}$ were available. Since publicizing $h_z^{-\log_h(g_r)}$ would render the scheme insecure, our idea is to use Waters signatures as a support for introducing extra randomizers in the exponent.

In the scheme, the u component of each signature can be seen as an aggregation of the one-time construction with a Waters signature $(h_z^{\log_h(g_r)} \cdot H_{\mathbb{G}}(\tau)^{-\rho}, h^{\rho})$ [248] on the tag τ.

Keygen(λ, n): given a security parameter λ and the dimension $n \in \mathbb{N}$ of the subspace to be signed, choose bilinear group $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^{\lambda}$. Then, conduct the following steps.

1. Choose $h \xleftarrow{\$} \mathbb{G}$ and $\alpha_z, \alpha_r, \beta_z \xleftarrow{\$} \mathbb{Z}_p$. Define $g_z = h^{\alpha_z}$, $g_r = h^{\alpha_r}$ and $h_z = h^{\beta_z}$.
2. For $i = 1$ to n, pick $\chi_i, \gamma_i, \delta_i \xleftarrow{\$} \mathbb{Z}_p$ and compute $g_i = g_z^{\chi_i} g_r^{\gamma_i}$, $h_i = h_z^{\chi_i} h^{\delta_i}$.
3. Choose a random vector $\overline{\mathbf{w}} = (w_0, w_1, \ldots, w_L) \xleftarrow{\$} \mathbb{G}^{L+1}$ and define a hash function $H_{\mathbb{G}} : \{0,1\}^L \to \mathbb{G}$ which maps the L-bit string $\tau = \tau[1] \ldots \tau[L] \in \{0,1\}^L$ to $H_{\mathbb{G}}(\tau) = w_0 \cdot \prod_{k=1}^{L} w_k^{\tau[k]}$.

The private key is $\mathsf{sk} = \big(h_z^{\alpha_r},\ \{\chi_i, \gamma_i, \delta_i\}_{i=1}^{n}\big)$ while the public key consists of

$$\mathsf{pk} = \big(g_z,\ g_r,\ h_z,\ h,\ \{g_i, h_i\}_{i=1}^{n},\ \overline{\mathbf{w}}\big) \in \mathbb{G}^{2n+4} \times \mathbb{G}^{L+1}.$$

Sign$(\mathsf{sk}, \tau, (M_1, \ldots, M_n))$: to sign $(M_1, \ldots, M_n) \in \mathbb{G}^n$ w.r.t. the file identifier τ using the private key $sk = \big(h_z^{\alpha_r},\ \{\chi_i, \gamma_i, \delta_i\}_{i=1}^{n}\big)$, choose $\theta, \rho \xleftarrow{\$} \mathbb{Z}_p$ and output $\sigma = (z, r, u, v)$, where

$$z = g_r^{\theta} \cdot \prod_{i=1}^{n} M_i^{-\chi_i} \qquad r = g_z^{-\theta} \cdot \prod_{i=1}^{n} M_i^{-\gamma_i}$$

$$u = (h_z^{\alpha_r})^{-\theta} \cdot \prod_{i=1}^{n} M_i^{-\delta_i} \cdot H_{\mathbb{G}}(\tau)^{-\rho} \qquad v = h^{\rho}$$

SignDerive$(\mathsf{pk}, \tau, \{(\omega_i, \sigma^{(i)})\}_{i=1}^{\ell})$: given pk, a file identifier τ and ℓ tuples $(\omega_i, \sigma^{(i)})$, parse $\sigma^{(i)}$ as $\sigma^{(i)} = \big(z_i, r_i, u_i, v_i\big) \in \mathbb{G}^4$ for $i = 1$ to ℓ. Then, choose $\rho' \xleftarrow{\$} \mathbb{Z}_p$ and compute and return $\sigma = (z, r, u, v)$, where $z = \prod_{i=1}^{\ell} z_i^{\omega_i}$, $r = \prod_{i=1}^{\ell} r_i^{\omega_i}$, $u = \prod_{i=1}^{\ell} u_i^{\omega_i} \cdot H_{\mathbb{G}}(\tau)^{-\rho'}$ and $v = \prod_{i=1}^{\ell} v_i^{\omega_i} \cdot h^{\rho'}$.

Verify$(\mathsf{pk}, \sigma, \tau, (M_1, \ldots, M_n))$: given $\sigma = (z, r, u, v) \in \mathbb{G}^4$, a file identifier τ and $(M_1, \ldots, M_n)$, return 1 if and only if $(M_1, \ldots, M_n) \neq (1_{\mathbb{G}}, \ldots, 1_{\mathbb{G}})$ and (z, r, u, v) satisfy

$$\begin{aligned} 1_{\mathbb{G}_T} &= e(g_z, z) \cdot e(g_r, r) \cdot \prod_{i=1}^{n} e(g_i, M_i), \qquad (3.2) \\ 1_{\mathbb{G}_T} &= e(h_z, z) \cdot e(h, u) \cdot e(H_{\mathbb{G}}(\tau), v) \cdot \prod_{i=1}^{n} e(h_i, M_i). \end{aligned}$$

The security of the scheme against *non-independent* Type I adversaries is proved under the SDP assumption. In the case of Type II forgeries, we need to assume the adversary to be independent because, at some point, the simulator is only able to compute a signature for a unique value[2] of θ.

Theorem 4 ([177]). *The scheme is unforgeable against independent adversaries if the SDP assumption holds in* $(\mathbb{G}, \mathbb{G}_T)$. *Moreover, the scheme is secure against non-independent Type I adversaries.*

Since the signature component u cannot be publicly randomized, the scheme does not have fully randomizable signatures. In Section 3.2.3, we describe a fully randomizable variant. In applications like non-malleable commitments to group elements, the above scheme is sufficient however.

[2] Note that this is not a problem since the signer can derive θ as a pseudorandom function of τ and $(M_1, \ldots, M_n)$ to make sure that a given vector is always signed using the same θ.

3.2.3 A Fully Randomizable Construction

We show that our scheme of Section 3.2.2 can be modified so as to become *strongly* context-hiding in the sense of [11, 25]. Namely, signatures produced by the SignDerive algorithm should be statistically indistinguishable from signatures freshly generated by Sign, even when the original signatures are given.

The difficulty is that, in the scheme of Section 3.2.2, we cannot re-randomize the underlying θ without knowing $h_z^{\alpha_r}$. To address this problem, it is tempting to include in each signature a randomization component of the form $(h_z^{\alpha_r} \cdot H_{\mathbb{G}}(\tau)^{-\zeta}, h^{\zeta})$, for some $\zeta \in \mathbb{Z}_p$, which can be seen as a signature on the vector $(1_{\mathbb{G}}, \ldots, 1_{\mathbb{G}})$. Unfortunately, the security proof ceases to go through as the reduction finds itself unable to generate a well-formed pair $(h_z^{\alpha_r} \cdot H_{\mathbb{G}}(\tau)^{-\zeta}, h^{\zeta})$ at some step of its interaction with the adversary. Our solution actually consists in committing to the signature components that cannot be re-randomized and provide evidence that committed group elements satisfy the verification equations. This is achieved using Groth-Sahai non-interactive arguments on a perfectly NIWI Groth-Sahai CRS, as in the linearly homomorphic construction of Attrapadung *et al.* [26]. A slight difference with [26], however, is that signature components $(H_{\mathbb{G}}(\tau)^{-\rho}, h^{-\rho})$ are no longer used and replaced by the technique of Malkin *et al.* [196], which yields slightly shorter signatures.

In the following notations, for each $h \in \mathbb{G}$ and any vector $\mathbf{g} = (g_1, g_2, g_3) \in \mathbb{G}^3$, we denote by $E(h, \mathbf{g})$ the vector $(e(h, g_1), e(h, g_2), e(h, g_3)) \in \mathbb{G}_T^3$.

Keygen(λ, n): given a security parameter λ and the dimension $n \in \mathbb{N}$ of the subspace to be signed, choose bilinear group $(\mathbb{G}, \mathbb{G}_T)$ of order $p > 2^{\lambda}$. Then, do the following.

1. Choose $h \xleftarrow{\$} \mathbb{G}$ and $\alpha_z, \alpha_r, \beta_z, \xleftarrow{\$} \mathbb{Z}_p$. Define $g_z = h^{\alpha_z}$, $g_r = h^{\alpha_r}$ and $h_z = h^{\beta_z}$.
2. For $i = 1$ to n, pick $\chi_i, \gamma_i, \delta_i \xleftarrow{\$} \mathbb{Z}_p$ and compute $g_i = g_z^{\chi_i} \cdot g_r^{\gamma_i}$, $h_i = h_z^{\chi_i} \cdot h^{\delta_i}$.
3. Generate $L+1$ Groth-Sahai CRSes by choosing $f_1, f_2 \xleftarrow{\$} \mathbb{G}$ and defining vectors $\mathbf{f_1} = (f_1, 1, g) \in \mathbb{G}^3$, $\mathbf{f_2} = (1, f_2, g) \in \mathbb{G}^3$ and $\mathbf{f}_{3,i} \xleftarrow{\$} \mathbb{G}^3$, for each $i \in \{0, \ldots, L\}$.

The private key is $\mathsf{sk} = \big(h_z^{\alpha_r},\ \{\chi_i, \gamma_i, \delta_i\}_{i=1}^n\big)$ while the public key consists of

$$\mathsf{pk} = \Big(g_z,\ g_r,\ h_z,\ h,\ \{g_i, h_i\}_{i=1}^n,\ \mathbf{f} = \big(\mathbf{f}_1, \mathbf{f}_2, \{\mathbf{f}_{3,i}\}_{i=0}^L\big)\Big).$$

Sign$(\mathsf{sk}, \tau, (M_1, \ldots, M_n))$: to sign a vector $(M_1, \ldots, M_n) \in \mathbb{G}^n$ using $\mathsf{sk} = \big(h_z^{\alpha_r},\ \{\chi_i, \gamma_i, \delta_i\}_{i=1}^n\big)$ with the file identifier τ, conduct the following steps.

1. Choose $\theta \xleftarrow{\$} \mathbb{Z}_p$ and compute
$$z = g_r^{\theta} \cdot \prod_{i=1}^n M_i^{-\chi_i} \qquad r = g_z^{-\theta} \cdot \prod_{i=1}^n M_i^{-\gamma_i} \qquad u = h_z^{-\theta \cdot \alpha_r} \cdot \prod_{i=1}^n M_i^{-\delta_i}$$
2. Using the bits $\tau[1] \ldots \tau[L]$ of $\tau \in \{0,1\}^L$, define the vector $\mathbf{f}_\tau = \mathbf{f}_{3,0} \cdot \prod_{i=1}^L \mathbf{f}_{3,i}^{\tau[i]}$ so as to assemble a Groth-Sahai CRS $\mathbf{f}_\tau = (\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_\tau)$.

3. Using $\mathbf{f}_\tau$, compute Groth-Sahai commitments

$$\begin{aligned}\mathbf{C}_z &= (1_{\mathbb{G}}, 1_{\mathbb{G}}, z) \cdot \mathbf{f_1}^{\nu_{z,1}} \cdot \mathbf{f_2}^{\nu_{z,2}} \cdot \mathbf{f}_\tau^{\nu_{z,3}}, \\ \mathbf{C}_r &= (1_{\mathbb{G}}, 1_{\mathbb{G}}, r) \cdot \mathbf{f_1}^{\nu_{r,1}} \cdot \mathbf{f_2}^{\nu_{r,2}} \cdot \mathbf{f}_\tau^{\nu_{r,3}}, \\ \mathbf{C}_u &= (1_{\mathbb{G}}, 1_{\mathbb{G}}, u) \cdot \mathbf{f_1}^{\nu_{u,1}} \cdot \mathbf{f_2}^{\nu_{u,2}} \cdot \mathbf{f}_\tau^{\nu_{u,3}}\end{aligned}$$

to z, r and u, respectively. Then, generate NIWI proofs $\boldsymbol{\pi}_1 = (\pi_{1,1}, \pi_{1,2}, \pi_{1,3}) \in \mathbb{G}^3$ and $\boldsymbol{\pi}_2 = (\pi_{2,1}, \pi_{2,2}, \pi_{2,3}) \in \mathbb{G}^3$ that (z, r, u) satisfy the pairing-product equations $1_{\mathbb{G}_T} = e(g_z, z) \cdot e(g_r, r) \cdot \prod_{i=1}^n e(g_i, M_i)$ and $1_{\mathbb{G}_T} = e(h_z, z) \cdot e(h, u) \cdot \prod_{i=1}^n e(h_i, M_i)$. These proofs are obtained as

$$\begin{aligned}\boldsymbol{\pi}_1 &= (\pi_{1,1}, \pi_{1,2}, \pi_{1,3}) = \big(g_z^{-\nu_{z,1}} \cdot g_r^{-\nu_{r,1}},\ g_z^{-\nu_{z,2}} \cdot g_r^{-\nu_{r,2}},\ g_z^{-\nu_{z,3}} \cdot g_r^{-\nu_{r,3}}\big) \\ \boldsymbol{\pi}_2 &= (\pi_{2,1}, \pi_{2,2}, \pi_{2,3}) = \big(h_z^{-\nu_{z,1}} \cdot h^{-\nu_{u,1}},\ h_z^{-\nu_{z,2}} \cdot h^{-\nu_{u,2}},\ h_z^{-\nu_{z,3}} \cdot h^{-\nu_{u,3}}\big)\end{aligned}$$

and satisfy the verification equations

$$\prod_{i=1}^n E\big(g_i, (1_{\mathbb{G}}, 1_{\mathbb{G}}, M_i)\big)^{-1} = E\big(g_z, \mathbf{C}_z\big) \cdot E\big(g_r, \mathbf{C}_r\big) \cdot E(\pi_{1,1}, \mathbf{f}_1) \cdot E(\pi_{1,2}, \mathbf{f}_2) \cdot E(\pi_{1,3}, \mathbf{f}_\tau) \tag{3.3}$$

$$\prod_{i=1}^n E\big(h_i, (1_{\mathbb{G}}, 1_{\mathbb{G}}, M_i)\big)^{-1} = E\big(h_z, \mathbf{C}_z\big) \cdot E\big(h, \mathbf{C}_u\big) \cdot E(\pi_{2,1}, \mathbf{f}_1) \cdot E(\pi_{2,2}, \mathbf{f}_2) \cdot E(\pi_{2,3}, \mathbf{f}_\tau).$$

The signature consists of

$$\sigma = (\mathbf{C}_z, \mathbf{C}_r, \mathbf{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2) \in \mathbb{G}^{15}. \tag{3.4}$$

SignDerive$(\mathsf{pk}, \tau, \{(\omega_i, \sigma^{(i)})\}_{i=1}^{\ell})$: given pk, a file identifier τ and ℓ tuples $(\omega_i, \sigma^{(i)})$, parse each signature $\sigma^{(i)}$ as a tuple of the form $\sigma^{(i)} = (\mathbf{C}_{z,i}, \mathbf{C}_{r,i}, \mathbf{C}_{u,i}, \boldsymbol{\pi}_{1,i}, \boldsymbol{\pi}_{2,i}) \in \mathbb{G}^{15}$ for $i = 1$ to ℓ. Otherwise, the derivation process proceeds in two steps.

1. Compute

$$\begin{aligned}\mathbf{C}_z &= \prod_{i=1}^{\ell} \mathbf{C}_{z,i}^{\omega_i} \qquad \mathbf{C}_r = \prod_{i=1}^{\ell} \mathbf{C}_{r,i}^{\omega_i} \qquad \mathbf{C}_u = \prod_{i=1}^{\ell} \mathbf{C}_{u,i}^{\omega_i} \\ \boldsymbol{\pi}_1 &= \prod_{i=1}^{\ell} \boldsymbol{\pi}_{1,i}^{\omega_i} \qquad \boldsymbol{\pi}_2 = \prod_{i=1}^{\ell} \boldsymbol{\pi}_{2,i}^{\omega_i}\end{aligned}$$

2. Re-randomize the above commitments and proofs using their homomorphic property and return the re-randomized version $\sigma = (\mathbf{C}_z, \mathbf{C}_r, \mathbf{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$.

Verify$(\mathsf{pk}, \sigma, \tau, (M_1, \ldots, M_n))$: given a pair $(\tau, (M_1, \ldots, M_n))$ and a purported signature σ parse the latter as $(\mathbf{C}_z, \mathbf{C}_r, \mathbf{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$. Then, return 1 if and only if it holds that $(M_1, \ldots, M_n) \neq (1_{\mathbb{G}}, \ldots, 1_{\mathbb{G}})$ and equations (3.3) are satisfied.

We believe this construction to be of interest even if we disregard its structure-preserving property. Indeed, if we compare it with the only known completely context-hiding linearly homomorphic signature in the standard model [26], its signatures are shorter by one group element. Moreover, we can prove the security under the sole DLIN assumption whereas the scheme of [26] requires an additional assumption.

The scheme is clearly completely context hiding because signatures only consist of perfectly randomizable commitments and NIWI arguments.

As for the unforgeability of the scheme, the proof of the following theorem is along the lines of [196, Theorem 5]. However, we can only prove unforgeability in a weaker sense as we need to assume that the adversary is targeting. Namely, in the case of Type II attacks, the adversary must also output a proof that it actually broke the security of the scheme and that its vector $\mathbf{M}^\star = (M_1^\star, \dots, M_n^\star) \in \mathbb{G}^n$ is indeed independent of the vectors for which it obtained signatures for the target tag $\tau^\star$.

If $\{\mathbf{M}_i = (M_{i,1}, \dots, M_{i,n})\}_{i=1}^m$ denote the linearly independent vectors that were signed for $\tau^\star$, the adversary could simply output a vector $\mathbf{W} = (W_1, \dots, W_n) \in \mathbb{G}^n$ such that $\prod_{j=1}^n e(M_j^\star, W_j) \neq 1_{\mathbb{G}_T}$ and $\prod_{j=1}^n e(M_{i,j}, W_j) = 1_{\mathbb{G}_T}$ for each $i \in \{1, \dots, m\}$. The latter test guarantees that the adversary's output is a non-trivial Type II forgery.

Theorem 5 ([177]). *The above scheme provides unforgeability against independent targeting adversaries if the DLIN assumption holds in* $\mathbb{G}$.

3.2.4 Application to Verifiable Computation on Encrypted Data

Linearly homomorphic schemes are known (see, *e. g.*, [11]) to provide verifiable computation mechanisms for outsourced data. Suppose that a user has a dataset consisting of n samples $s_1, \dots, s_n \in \mathbb{Z}_p$. The dataset can be encoded as vectors $\mathbf{v}_i = (\mathbf{e}_i | s_i) \in \mathbb{Z}_p^{n+1}$, where $\mathbf{e}_i \in \mathbb{Z}_p^n$ denotes the i-th unit vector for each $i \in \{1, \dots, n\}$. The user then assigns a file identifier τ to $\{\mathbf{v}_i\}_{i=1}^n$, computes signatures $\sigma_i \leftarrow \mathsf{Sign}(\mathsf{sk}, \tau, \mathbf{v}_i)$ on the resulting vectors and stores $\{(\mathbf{v}_i, \sigma_i)\}_{i=1}^n$ at the server. When requested, the server can then evaluate a sum $s = \sum_{i=1}^n s_i$ and provide evidence that the latter computation is correct by deriving a signature on the vector $(1, 1, \dots, 1, s) \in \mathbb{Z}_p^{n+1}$. Unless the server is able to forge a signature for a vector outside the span of $\{\mathbf{v}_i\}_{i=1}^n$, it is unable to fool the user. The above method readily extends to authenticate weighted sums or Fourier transforms.

One disadvantage of the above method is that it requires the server to retain the dataset $\{s_i\}_{i=1}^n$ in the clear. Using LHSPS schemes, the user can apply the above technique on encrypted samples using the Boneh-Boyen-Shacham (BBS) cryptosystem [44].

The BBS cryptosystem involves a public key $(g, \tilde{g}, f = g^x, h = g^y) \in_R \mathbb{G}^4$, where $(x, y) \in \mathbb{Z}_p^2$ is the private key. The user (or anyone else knowing his public key) can first encrypt his samples $\{s_i\}_{i=1}^n$ by computing BBS encryptions $(C_{1,i}, C_{2,i}, C_{3,i}) = (f^{r_i}, h^{t_i}, \tilde{g}^{s_i} \cdot g^{r_i + t_i})$, with $r_i, t_i \xleftarrow{\$} \mathbb{Z}_p$, for each $i \in \{1, \dots, n\}$. If the user holds a LHSPS key pair for vectors of dimension $n+3$, he can generate n signatures on vectors $((C_{1,i}, C_{2,i}, C_{3,i}) | \mathbf{E}_i) \in \mathbb{G}^{n+3}$, where $\mathbf{E}_i = (1_\mathbb{G}, \dots, 1_\mathbb{G}, g, 1_\mathbb{G}, \dots, 1_\mathbb{G}) = g^{\mathbf{e}_i}$ for each $i \in \{1, \dots, n\}$, using the scheme of Section 3.2.2. The vectors $\{((C_{1,i}, C_{2,i}, C_{3,i}) | \mathbf{E}_i)\}_{i=1}^n$ and their signatures $\{(z_i, r_i, u_i, v_i)\}_{i=1}^n$ are then archived in the cloud in such a way that the server can publicly derive a signature on the vector $\big(f^{\sum_i r_i}, h^{\sum_i t_i}, \tilde{g}^{\sum_i s_i} \cdot g^{\sum_i (r_i + t_i)}, g, g, \dots, g\big) \in \mathbb{G}^{n+3}$ in order to convince the client that the encrypted sum was correctly computed. Using his private key (x, y), the client can then retrieve the sum $\sum_i s_i$ as long as it remains in a sufficiently small range.

The interest of the above solution lies in that the client can dispense with the need for storing the $O(n)$-size public key of his linearly homomorphic signature. Indeed, he can simply retain the random seed that was used to generate pk and re-compute private key elements $\{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^n$ whenever he wants to verify the server's response. In this case, the verification equations (3.2) become

$$1_{\mathbb{G}_T} = e(g_z, z \cdot \prod_{i=1}^n M_i^{\chi_i}) \cdot e(g_r, r \cdot \prod_{i=1}^n M_i^{\gamma_i}) = e(h_z, z \cdot \prod_{i=1}^n M_i^{\chi_i}) \cdot e(h, u \cdot \prod_{i=1}^n M_i^{\delta_i}) \cdot e(H_{\mathbb{G}}(\tau), v),$$

so that the client only has to compute $O(1)$ pairings. Moreover, the client does not have to determine an upper bound on the size of his dataset when generating his public key. Initially, he only needs to generate $\{(g_j, h_j)\}_{j=1}^3$. When the i-th ciphertext $(C_{1,i}, C_{2,i}, C_{3,i})$ has to be stored, the client derives $(\chi_{i+3}, \gamma_{i+3}, \delta_{i+3})$ and (g_{i+3}, h_{i+3}) by applying a PRF to the index i. This will be sufficient to sign vectors of the form $((C_{1,i}, C_{2,i}, C_{3,i})|\mathbf{E}_i)$.

Complete and security models for "verifiable computation on encrypted data" are beyond the scope of this work. Here, they would naturally combine the properties of secure homomorphic encryption and authenticated computing. It should be intuitively clear that a malicious server cannot trick a client into accepting an incorrect result (i.e., one which differs from the actual defined linear function it is supposed to compute over the defined signed ciphertext inputs) without defeating the security of the underlying homomorphic signature.

3.3 Non-Malleable Trapdoor Commitments to Group Elements from Linearly Homomorphic Structure-Preserving Signatures

This section shows that, under a certain mild condition (fulfilled by our constructions), LH-SPS imply length-reducing non-malleable structure-preserving commitments to vectors of group elements.

As a result, we obtain the first length-reducing non-malleable structure-preserving trapdoor commitment. Our scheme is not *strictly*[3] structure-preserving (according to the terminology of [7]) because the commitment string lives in $\mathbb{G}_T$ rather than $\mathbb{G}$. Still, openings only consist of elements in $\mathbb{G}$, which makes it possible to generate efficient NIWI proofs that committed group elements satisfy certain properties. To our knowledge, the only known non-malleable commitment schemes whose openings only consist of group elements were described by Fischlin *et al.* [108]. However, these constructions cannot be length-reducing as they achieve universal composability [70, 71].

Our schemes are obtained by first constructing simulation-sound trapdoor commitments (SSTC) [116, 195] to group elements. SSTC schemes were first suggested by Garay, MacKenzie and Yang [116] as a tool for constructing universally composable zero-knowledge proofs [70]. MacKenzie and Yang subsequently gave a simplified security definition which suffices to provide non-malleability with respect to opening in the sense of the definition of re-usable non-malleable commitments [94].

In a SSTC, each commitment is labeled with a tag. The definition of [195] requires that, even if the adversary can see equivocations of commitments to possibly distinct messages for several tags $tag_1, \ldots, tag_q$, it will not be able to break the binding property for a new tag $tag \notin \{tag_1, \ldots, tag_q\}$.

[3] We recall that strictly structure-preserving commitments cannot be length-reducing, as shown by Abe *et al.* [7], so that our scheme is essentially the best we can hope for if we aim at short commitment stings.

Definition 7 ([195]). *A simulation-sound trapdoor commitment (SSTC)* $(\mathsf{Setup}, \mathsf{Com}, \mathsf{FakeCom}, \mathsf{FakeOpen}, \mathsf{Verify})$ *is a tuple where* $(\mathsf{Setup}, \mathsf{Com}, \mathsf{Verify})$ *forms a non-interactive commitment scheme and* $(\mathsf{FakeCom}, \mathsf{FakeOpen})$ *are PPT algorithms with the following properties*

Trapdoor: *for any tag and any message* Msg*, the following distributions are computationally indistinguishable:*

$$D_{fake} := \{(pk, tk) \leftarrow \mathsf{Setup}(\lambda);\ (\widetilde{\mathsf{com}}, \mathsf{aux}) \leftarrow \mathsf{FakeCom}(pk, tk, tag);$$
$$\widetilde{\mathsf{dec}} \leftarrow \mathsf{FakeOpen}(\mathsf{aux}, tk, \widetilde{\mathsf{com}}, \mathsf{Msg}) : (pk, tag, \mathsf{Msg}, \widetilde{\mathsf{com}}, \widetilde{\mathsf{dec}})\}$$

$$D_{real} := \{(pk, tk) \leftarrow \mathsf{Setup}(\lambda);\ (\mathsf{com}, \mathsf{dec}) \leftarrow \mathsf{Com}(pk, tag, \mathsf{Msg}) : (pk, tag, \mathsf{Msg}, \mathsf{com}, \mathsf{dec})\}$$

Simulation-sound binding: *for any PPT adversary $\mathcal{A}$, the following probability is negligible*

$$\Pr[(pk, tk) \leftarrow \mathsf{Setup}(\lambda);\ (\mathsf{com}, tag, \mathsf{Msg}_1, \mathsf{Msg}_2, \mathsf{dec}_1, \mathsf{dec}_2) \leftarrow \mathcal{A}^{\mathcal{O}_{tk,pk}}(pk) : \mathsf{Msg}_1 \neq \mathsf{Msg}_2$$
$$\wedge\ \mathsf{Verify}(pk, tag, \mathsf{Msg}_1, \mathsf{com}, \mathsf{dec}_1) = \mathsf{Verify}(pk, tag, \mathsf{Msg}_2, \mathsf{com}, \mathsf{dec}_2) = 1 \wedge tag \notin Q],$$

where $\mathcal{O}_{tk,pk}$ is an oracle that maintains an initially empty set Q and operates as follows:

- *On input* (commit, tag)*, it runs* $(\widetilde{\mathsf{com}}, \mathsf{aux}) \leftarrow \mathsf{FakeCom}(pk, tk, tag)$*, stores the triple* $(\widetilde{\mathsf{com}}, tag, \mathsf{aux})$*, returns* $\widetilde{\mathsf{com}}$*.*
- *On input* $(\mathsf{decommit}, \widetilde{\mathsf{com}}, \mathsf{Msg})$*: if a tuple* $(\widetilde{\mathsf{com}}, tag, \mathsf{aux})$ *was previously stored, it computes* $\widetilde{\mathsf{dec}} \leftarrow \mathsf{FakeOpen}(\mathsf{aux}, tk, tag, \widetilde{\mathsf{com}}, \mathsf{Msg})$*, adds tag in Q and returns* $\widetilde{\mathsf{dec}}$*. Otherwise, $\mathcal{O}_{tk,pk}$ returns $\perp$.*

While our SSTC to group elements will be proved secure in the above sense, a *non-adaptive* flavor of simulation-sound binding security is sufficient for the construction of non-malleable commitments. Indeed, Gennaro used [118] such a relaxed notion to achieve non-malleability from similar-looking multi-trapdoor commitments. In the non-adaptive notion, the adversary has to choose the set of tags $tag_1, \ldots, tag_\ell$ for which it wants to query the $\mathcal{O}_{tk,pk}$ oracle before seeing the public key pk.

3.3.1 Template of Linearly Homomorphic SPS Scheme

We first remark that *any* constant-size linearly homomorphic structure-preserving signature necessarily complies with the template below. Indeed, in order to have a linear homomorphism, each verification equation necessarily computes a product of pairings which should equal $1_{\mathbb{G}_T}$ in a valid signature. In each pairing of the product, one of the arguments must be a message or signature component while the second argument is either part of the public key or an encoding of the file identifier.

For simplicity, the template is described in terms of symmetric pairings but generalizations to asymmetric configurations are possible.

Keygen(λ, n): given λ and the dimension $n \in \mathbb{N}$ of the vectors to be signed, choose constants n_z, n_v, m. Among these, n_z and n_v will determine the signature length while

m will be the number of verification equations. Then, choose $\{F_{j,\mu}\}_{j\in\{1,\dots,m\},\mu\in\{1,\dots,n_z\}}$, $\{G_{j,i}\}_{i\in\{1,\dots,n\},\ j\in\{j,\dots,m\}}$ in the group $\mathbb{G}$. The public key is

$$\mathsf{pk} = \big(\{F_{j,\mu}\}_{j\in\{1,\dots,m\},\mu\in\{1,\dots,n_z\}}, \{G_{j,i}\}_{i\in\{1,\dots,n\},\ j\in\{j,\dots,m\}}\big)$$

while sk consists of information about the representation of public elements w.r.t. specific bases.

Sign($\mathsf{sk}, \tau, (M_1, \dots, M_n)$): Outputs a tuple $\sigma = (Z_1, \dots, Z_{n_z}, V_1, \dots, V_{n_v}) \in \mathbb{G}^{n_z+n_v}$.

SignDerive($\mathsf{pk}, \tau, \{(\omega_i, \sigma^{(i)})\}_{i=1}^{\ell}$): parses each $\sigma^{(i)}$ as $(Z_1^{(i)}, \dots, Z_{n_z}^{(i)}, V_1^{(i)}, \dots, V_{n_v}^{(i)})$ and computes

$$Z_\mu = \prod_{i=1}^{\ell} Z_\mu^{(i)\ \omega_i} \qquad V_\nu = \prod_{i=1}^{\ell} V_\nu^{(i)\ \omega_i} \qquad \mu \in \{1, \dots, n_z\},\ \nu \in \{1, \dots, n_v\}.$$

After a possible extra re-randomization step, it outputs $(Z_1, \dots, Z_{n_z}, V_1, \dots, V_{n_v})$.

Verify($\mathsf{pk}, \sigma, \tau, (M_1, \dots, M_n)$): given a signature $\sigma = (Z_1, \dots, Z_{n_z}, V_1, \dots, V_{n_v}) \in \mathbb{G}^{n_z+n_v}$, a tag τ and $(M_1, \dots, M_n)$, return 0 if $(M_1, \dots, M_n) = (1_{\mathbb{G}}, \dots, 1_{\mathbb{G}})$. Otherwise, do the following.

1. For each $j \in \{1, \dots, m\}$ and $\nu \in \{1, \dots, n_v\}$, compute one-to-one[4] encodings $T_{j,\nu} \in \mathbb{G}$ of the tag τ as a group element.
2. Return 1 if and only if $c_j = 1_{\mathbb{G}_T}$ for $j = 1$ to m, where

$$c_j = \prod_{\mu=1}^{n_z} e(F_{j,\mu}, Z_\mu) \cdot \prod_{\nu=1}^{n_v} e(T_{j,\nu}, V_\nu) \cdot \prod_{i=1}^{n} e(G_{j,i}, M_i) \qquad j \in \{1, \dots, m\}. \tag{3.5}$$

In the following, we say that a linearly homomorphic SPS is *regular* if, for each file identifier τ, any non-trivial vector $(M_1, \dots, M_n) \neq (1_{\mathbb{G}}, \dots, 1_{\mathbb{G}})$ has a valid signature.

3.3.2 Construction of Simulation-Sound Structure-Preserving Trapdoor Commitments

Let $\Pi^{\mathsf{SPS}} = (\mathsf{Keygen}, \mathsf{Sign}, \mathsf{SignDerive}, \mathsf{Verify})$ be a linearly homomorphic SPS. We construct a simulation-sound trapdoor commitment as follows.

SSTC.Setup(λ, n): given the desired dimension $n \in \mathbb{N}$ of committed vectors, choose public parameters pp for the linearly homomorphic SPS scheme. Then, run $\Pi^{\mathsf{SPS}}.\mathsf{Keygen}(\lambda, n)$ to obtain a public key $\mathsf{pk} = \big(\{F_{j,\mu}\}_{j\in\{1,\dots,m\},\mu\in\{1,\dots,n_z\}}, \{G_{j,i}\}_{i\in\{1,\dots,n\},\ j\in\{j,\dots,m\}}\big)$, for some constants n_z, n_v, m, and a sk. The commitment key is $pk = \mathsf{pk}$ and the trapdoor tk consists of sk. Note that the public key defines a signature space $\mathbb{G}^{n_z+n_v}$, for constants n_z and n_v.

[4] This condition can be relaxed to have collision-resistant deterministic encodings. Here, we assume injectivity for simplicity.

SSTC.Com$(pk, tag, (M_1, \ldots, M_n))$**:** to commit to $(M_1, \ldots, M_n) \in \mathbb{G}^n$ with respect to the tag $tag = \tau$, choose $\big(Z_1, \ldots, Z_{n_z}, V_1, \ldots, V_{n_v}\big) \xleftarrow{\$} \mathbb{G}^{n_z+n_v}$ in the signature space. Then, run step 1 of the verification algorithm and evaluate the right-hand-side member of (3.5). Namely, compute

$$c_j \;=\; \prod_{\mu=1}^{n_z} e(F_{j,\mu}, Z_\mu) \cdot \prod_{\nu=1}^{n_v} e(T_{j,\nu}, V_\nu) \cdot \prod_{i=1}^{n} e(G_{j,i}, M_i) \qquad j \in \{1, \ldots, m\} \tag{3.6}$$

where $\{T_{j,\nu}\}_{j,\nu}$ form an injective encoding of $tag = \tau$ as a set of group elements. The commitment string is defined to be $\mathsf{com} = (c_1, \ldots, c_m)$ whereas the decommitment consists of $\mathsf{dec} = \big(Z_1, \ldots, Z_{n_z}, V_1, \ldots, V_{n_v}\big)$.

SSTC.FakeCom(pk, tk, tag)**:** proceeds like SSTC.Com with $(\hat{M}_1, \ldots, \hat{M}_n) \xleftarrow{\$} \mathbb{G}^n$. If $(\hat{\mathsf{com}}, \hat{\mathsf{dec}})$ denotes the resulting pair, the algorithm outputs $\widetilde{\mathsf{com}} = \hat{\mathsf{com}}$ and the auxiliary information aux, which consists of the pair $\mathsf{aux} = ((\hat{M}_1, \ldots, \hat{M}_n), \hat{\mathsf{dec}})$ for $tag = \tau$.

SSTC.FakeOpen$(\mathsf{aux}, tk, tag, \widetilde{\mathsf{com}}, (M_1, \ldots, M_n))$**:** the algorithm parses $\widetilde{com}$ as $(\tilde{c}_1, \ldots, \tilde{c}_m)$ and aux as $\big((\hat{M}_1, \ldots, \hat{M}_n), (\hat{Z}_1, \ldots, \hat{Z}_{n_z}, \hat{V}_1, \ldots, \hat{V}_{n_v})\big)$. It first generates a homomorphic signature on $(M_1/\hat{M}_1, \ldots, M_n/\hat{M}_n)$ for the tag $tag = \tau$. Namely, using $tk = \mathsf{sk}$, compute $\sigma' = (Z'_1, \ldots, Z'_{n_z}, V'_1, \ldots, V'_{n_v}) \leftarrow \Pi^{\mathsf{SPS}}.\mathsf{Sign}\big(\mathsf{sk}, \tau, (M_1/\hat{M}_n, \ldots, M_n/\hat{M}_n)\big)$. Since σ' is a valid signature and $\mathsf{aux} = \big((\hat{M}_1, \ldots, \hat{M}_n), (\hat{Z}_1, \ldots, \hat{Z}_{n_z}, \hat{V}_1, \ldots, \hat{V}_{n_v})\big)$ satisfies

$$\tilde{c}_j \;=\; \prod_{\mu=1}^{n_z} e(F_{j,\mu}, \hat{Z}_\mu) \cdot \prod_{\nu=1}^{n_v} e(T_{j,\nu}, \hat{V}_\nu) \cdot \prod_{i=1}^{n} e(G_{j,i}, \hat{M}_i) \qquad j \in \{1, \ldots, m\}, \tag{3.7}$$

the algorithm can run $(\tilde{Z}_1, \ldots, \tilde{Z}_{n_z}, \tilde{V}_1, \ldots, \tilde{V}_{n_v}) \leftarrow \mathsf{SignDerive}(\mathsf{pk}, \tau, \{(1, \sigma'), (1, \hat{\sigma})\})$, where $\hat{\sigma} = (\hat{Z}_1, \ldots, \hat{Z}_{n_z}, \hat{V}_1, \ldots, \hat{V}_{n_v})$, and output $\widetilde{\mathsf{dec}} = (\tilde{Z}_1, \ldots, \tilde{Z}_{n_z}, \tilde{V}_1, \ldots, \tilde{V}_{n_v})$ which is a valid de-commitment to the vector $(M_1, \ldots, M_n)$ with respect to $tag = \tau$.

SSTC.Verify$(pk, tag, (M_1, \ldots, M_n), \mathsf{com}, \mathsf{dec})$**:** parse com as $(c_1, \ldots, c_m) \in \mathbb{G}_T^m$ and the de-commitment dec as $\big(Z_1, \ldots, Z_{n_z}, V_1, \ldots, V_{n_v}\big) \in \mathbb{G}^{n_z+n_v}$ (if these values do not parse properly, return 0). Then, compute a one-to-one encoding $\{T_{j,\nu}\}_{j,\nu}$ of $tag = \tau$. Return 1 if relations (3.6) hold and 0 otherwise.

In the full version of [177], we generalize the above construction so as to build simulation-sound trapdoor commitment to vectors from any linearly homomorphic signature that fits a certain template. This template captures essentially all known pairing-based constructions, including LHSPS schemes. As a result, we obtain a modular construction of constant-size non-malleable commitment to vectors which preserves the feasibility of efficiently proving properties about committed values. In particular, our generalized construction can be instantiated using the CDH-based (non-structure-preserving) linearly homomorphic signature of Attrapadung, Libert and Peters [25]. Unlike the CDH-based simulation-sound commitment of Fujisaki [114], our realization is non-interactive and allows committing to vectors with a constant-size commitment string. Unlike the solution consisting in committing to a short string obtained by hashing the vector, our solution allows the sender to prove properties (using Σ protocols or Groth-Sahai proofs) about committed vectors in an efficient way.

For vectors of dimension $n = 1$, we obtain a simplification of existing multi-trapdoor (or identity-based) trapdoor commitments [100, 214] based on Waters signatures. Our generalized construction of simulation-sound commitments [177] can also be instantiated under the Strong Diffie-Hellman assumption using the homomorphic signature of Catalano *et al.* [78]. For vectors of dimension 1, the obtained non-malleable commitment is a variant of the one of [118, Section 4.2].

Theorem 6 ([177]). *Assuming that the underlying linearly homomorphic SPS is regular and secure against non-independent Type I adversaries, the above construction is a simulation-sound trapdoor commitment to group elements.*

A standard technique (see, e.g., [116, 118]) to build a re-usable and non-interactive non-malleable commitment (assuming a CRS) from a SSTC scheme is as follows. To commit to Msg, the sender generates a key-pair $(\mathsf{VK}, \mathsf{SK})$ for a one-time signature and generates $(\mathsf{com}, \mathsf{dec}) \leftarrow \mathsf{SSTC.Commit}(pk, \mathsf{VK}, \mathsf{MSg})$ using VK as a tag. The non-malleable commitment string is the pair $(\mathsf{com}, \mathsf{VK})$ and the opening is given by (dec, σ), where σ is a one-time signature on com, so that the receiver additionally checks the validity of σ. This construction is known to provide independence [93, 121] and thus non-malleability with respect to opening, as proved in [93, 121].

In our setting, we cannot compute σ as a signature of com, as it consists of $\mathbb{G}_T$ elements. However, we can rather sign the pair $(\mathsf{Msg}, \mathsf{dec})$ — whose components live in $\mathbb{G}$ — as long as it uniquely determines com. To this end, we can use the one-time structure-preserving of [6, Appendix C.1] since it allows signing messages of arbitrary length using a constant-size one-time public key. Like our scheme of Section 3.2.2, it relies on the SDP assumption and thus yields a non-malleable commitment based on this sole assumption. Alternatively, we can move σ in the commitment string (which thus consists of $(\mathsf{com}, \mathsf{VK}, \sigma)$), in which case the one-time signature does not need to be structure-preserving but it has to be strongly unforgeable (as can be observed from the definition of independent commitments [93]) while the standard notion of unforgeability suffices in the former case.

3.4 (Constant-Size) Simulation-Sound Quasi-Adaptive NIZK Arguments from LHSPS Schemes

Earlier sections showed that structure-preserving signatures with additive homomorphic properties have unexpected applications in the design of non-malleable structure-preserving commitments. In this section, we extend their range of applications and demonstrate that they can surprisingly be used (albeit non-generically) in the design of simulation-sound quasi-adaptive NIZK (QA-NIZK) proofs and chosen-ciphertext-secure cryptosystems.

Concretely, our one-time LHSPS scheme of Section 3.2.1 already allows showing membership of a $t \times n$ linear subspace (of rank $t < n$) using only 3 group elements under the SDP assumption. Moreover, we show how to extend this construction to get unbounded simulation-soundness while retaining *constant-size* proofs. The length of a proof does not depend on the number of equations or the number of variables, but only on the underlying assumption. Like those of [151], our proofs are computationally sound under standard assumptions. Somewhat surprisingly, they are even asymptotically shorter than random-oracle-based proofs derived from Σ-protocols.

Under the DLIN assumption, we obtain QA-NIZK arguments consisting of 15 group elements and a one-time signature with its verification key. As it turns out, it is also the first unbounded simulation-sound proof system that does not involve quadratic pairing product equations or a CCA2-secure encryption scheme. Efficiency comparisons show that we only need 20 group elements per proof where the best USS extension [62] of Groth-Sahai costs $6t + 2n + 52$ group elements. Under the k-linear assumption, the proof length becomes $O(k^2)$ and thus avoids any dependency on the subspace dimension.

For applications, like CCA2 security [208, 231], where only one-time simulation-soundness is needed, we further optimize our proof system and obtain a relatively simulation-sound QA-NIZK proof system, as defined in [150], with constant-size proofs. Under the DLIN assumption (resp. the k-linear assumption), we achieve relative simulation-soundness with only 4 (resp. $k + 2$) group elements!

As the first application of USS proofs, we construct a chosen-ciphertext-secure keyed-homomorphic encryption scheme with threshold decryption. Keyed-homomorphic encryption is a primitive, suggested by Emura *et al.* [104], where homomorphic ciphertext manipulations are only possible to a party holding a devoted evaluation key SK_h which, by itself, does not enable decryption. The scheme should provide IND-CCA2 security when the evaluation key is unavailable to the adversary and remain IND-CCA1 secure when SK_h is exposed. Other approaches to reconcile homomorphism and non-malleability were taken in [221, 222, 223, 51, 83] but they inevitably satisfy weaker security notions than adaptive chosen-ciphertext security [226]. The results of [104] showed that CCA2-security does not rule out homomorphicity when the capability to compute over encrypted data is restricted.

Emura *et al.* [104] gave realizations of CCA2-secure keyed-homomorphic schemes based on hash proof systems [90]. However, these do not readily enable threshold decryption – as would be desirable in voting protocols – since valid ciphertexts are not publicly recognizable, which makes it harder to prove CCA security in the threshold setting. Moreover, these solutions are not known to satisfy the strongest security definition of [104]. The reason is that this definition seemingly requires a form of unbounded simulation-soundness. Our QA-NIZK proofs fulfill this requirement and provide an efficient CCA2-secure threshold keyed-homomorphic system where ciphertexts are 65% shorter than in instantiations of the same high-level idea using previous simulation-sound proofs.

Using our relatively simulation-sound QA-NIZK proofs, we then build adaptively secure non-interactive threshold cryptosystems with CCA2 security and improved efficiency. The constructions of Libert and Yung [191] were improved by Escala *et al.* [105]. So far, the most efficient solution is obtained from the Jutla-Roy results [150, 151] via relatively sound proofs [150]. Using our relatively sound QA-NIZK proof system, we shorten ciphertexts by $\Theta(k)$ elements under the k-linear assumption.

3.4.1 Construction with Unbounded Simulation-Soundness

In the following, vectors are considered as row vectors. If $\mathbf{A} \in \mathbb{Z}_p^{t\times n}$ is a matrix, we denote by $g^{\mathbf{A}} \in \mathbb{G}^{t\times n}$ the matrix obtained by exponentiating g using the entries of $\mathbf{A}$.

We consider public parameters $\Gamma = (\mathbb{G}, \mathbb{G}_T, g)$ consisting of bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ with a generator $g \in \mathbb{G}$. Like [151], we will consider languages $\mathcal{L}_\rho = \{g^{\mathbf{x}\cdot\mathbf{A}} \in \mathbb{G}^n \mid \mathbf{x} \in \mathbb{Z}_p^t\}$ that are parametrized by $\rho = g^{\mathbf{A}} \in \mathbb{G}^{t\times n}$, where $\mathbf{A} \in \mathbb{Z}_q^{t\times n}$ is a $t \times n$ matrix of rank $t < n$.

As in [151], we assume that the distribution $\mathcal{D}_\Gamma$ is efficiently samplable: there exists a PPT algorithm which outputs a pair $(\rho, \mathbf{A})$ describing a relation R_ρ and its associated language $\mathcal{L}_\rho$ according to $\mathcal{D}_\Gamma$. One example of such a distribution is obtained by picking a uniform matrix $\mathbf{A} \overset{\$}{\leftarrow} \mathbb{Z}_p^{t\times n}$ – which has full rank with overwhelming probability – and setting $\rho = g^{\mathbf{A}}$.

Our construction builds on the homomorphic signature recalled in Section 3.2.3. Specifically, the language-dependent CRS ψ contains one-time linearly homomorphic signatures on the rows of the matrix $\rho \in \mathbb{G}^{t \times n}$. For each vector $\mathbf{v} \in \mathcal{L}_\rho$, the prover can use the witness $\mathbf{x} \in \mathbb{Z}_p^t$ to derive and prove knowledge of a one-time homomorphic signature (z, r, u) on $\mathbf{v}$. This signature (z, r, u) is already a QA-NIZK proof of membership but it does not provide simulation-soundness. To acquire this property, we follow [196] and generate a NIWI proof of knowledge of (z, r, u) for a Groth-Sahai CRS that depends on the verification key of an ordinary one-time signature. The latter's private key is used to sign the NIWI proof so as to prevent unwanted proof manipulations. Using the private key of the homomorphic one-time signature as a trapdoor, the simulator is also able to create proofs for vectors $\mathbf{v} \notin \mathcal{L}_\rho$. Due to the use of perfectly NIWI proofs, these fake proofs do not leak any more information about the simulation key than the CRS does. At the same time, the CRS can be prepared so that, with non-negligible probability, it becomes perfectly binding on an adversarially-generated proof, which allows extracting a non-trivial signature on a vector $\mathbf{v} \notin \mathcal{L}_\rho$.

Like [151], our QA-NIZK proof system $(\mathbb{K}_0, \mathbb{K}_1, \mathsf{P}, \mathsf{V})$ is a split CRS construction in that $\mathbb{K}_1$ can be divided into two algorithms $(\mathbb{K}_{10}, \mathbb{K}_{11})$. The first one $\mathbb{K}_{10}$ outputs some state information s and a first CRS $\mathbf{CRS}_2$ which is only used by the verifier and does not depend on the language $\mathcal{L}_\rho$. The second part $\mathbb{K}_{11}$ of $\mathbb{K}_1$ inputs the state information s and the output of Γ of $\mathbb{K}_0$ and outputs $\mathbf{CRS}_1$ which is only used by the prover.

$\mathbb{K}_0(\lambda)$**:** choose symmetric bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ with $g \xleftarrow{\$} \mathbb{G}$. Then, output $\Gamma = (\mathbb{G}, \mathbb{G}_T, g)$

The dimensions (t, n) of the matrix $\mathbf{A} \in \mathbb{Z}_p^{t \times n}$ can be either fixed or part of the language, so that t, n can be given as input to the CRS generation algorithm $\mathbb{K}_1$.

$\mathbb{K}_1(\Gamma, \rho)$**:** parse Γ as $(\mathbb{G}, \mathbb{G}_T, g)$ and ρ as a matrix $\rho = \big(G_{i,j}\big)_{1 \le i \le t,\ 1 \le j \le n} \in \mathbb{G}^{t \times n}$.

1. Generate a key pair $(\mathsf{pk}_{hsps}, \mathsf{sk}_{hsps})$ for the randomizable LHSPS of Section 3.2.3 to sign vectors of $\mathbb{G}^n$. Namely, choose $g_z, g_r, h_z, h_u \xleftarrow{\$} \mathbb{G}$ and do the following.

   a. For $i = 1$ to n, pick $\chi_i, \gamma_i, \delta_i \xleftarrow{\$} \mathbb{Z}_p$ and compute $g_i = g_z{}^{\chi_i} g_r{}^{\gamma_i}$ and $h_i = h_z{}^{\chi_i} h_u{}^{\delta_i}$.

   b. Generate $L + 1$ Groth-Sahai common reference strings, for some $L \in \mathsf{poly}(\lambda)$. To this end, choose $f_1, f_2 \xleftarrow{\$} \mathbb{G}$ and define the vectors $\mathbf{f}_1 = (f_1, 1, g) \in \mathbb{G}^3$, $\mathbf{f}_2 = (1, f_2, g) \in \mathbb{G}^3$. Then, pick $\mathbf{f}_{3,i} \xleftarrow{\$} \mathbb{G}^3$ for $i = 0$ to L.

   Let $\mathsf{sk}_{hsps} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^n$ be the private key and the matching public key is

$$\mathsf{pk}_{hsps} \quad = \quad \Big(g_z,\ g_r,\ h_z,\ h_u,\ \{(g_i, h_i)\}_{i=1}^n,\ \mathbf{f} = \big(\mathbf{f}_1, \mathbf{f}_2, \{\mathbf{f}_{3,i}\}_{i=0}^L\big) \Big).$$

2. Use sk_{hsps} to generate one-time linearly homomorphic signatures $\{(z_i, r_i, u_i)\}_{i=1}^t$ on the vectors $(G_{i1}, \dots, G_{in}) \in \mathbb{G}^n$ that form the rows of ρ. These are obtained as

$$(z_i, r_i, u_i) \quad = \quad \big(\prod_{j=1}^n G_{i,j}^{-\chi_j},\ \prod_{j=1}^n G_{i,j}^{-\gamma_j},\ \prod_{j=1}^n G_{i,j}^{-\delta_j}\big) \qquad \forall i \in \{1, \dots, t\}.$$

3. Choose a strongly unforgeable one-time signature $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ with verification keys consisting of L-bit strings.

4. The CRS $\psi = (\mathbf{CRS}_1, \mathbf{CRS}_2)$ consists of two parts which are defined as

$$\mathbf{CRS}_1 = \Big(\rho,\ \mathsf{pk}_{hsps},\ \{(z_i, r_i, u_i)\}_{i=1}^t,\ \Sigma\Big), \qquad \mathbf{CRS}_2 = \Big(\mathsf{pk}_{hsps},\ \Sigma\Big),$$

while the simulation trapdoor τ_{sim} is $\mathsf{sk}_{hsps} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^n$.

$\mathsf{P}(\Gamma, \psi, \mathbf{v}, x, \mathsf{lbl})$**:** given a vector $\mathbf{v} \in \mathbb{G}^n$ and a witness $\mathbf{x} = (x_1, \ldots, x_t) \in \mathbb{Z}_p^t$ such that $\mathbf{v} = g^{\mathbf{x} \cdot \mathbf{A}}$, generate a one-time signature key pair $(\mathsf{VK}, \mathsf{SK}) \leftarrow \mathcal{G}(\lambda)$ and do the following.

1. Using $\{(z_j, r_j, u_j)\}_{j=1}^t$, derive a one-time linearly homomorphic signature (z, r, u) on $\mathbf{v}$. Namely, compute $z = \prod_{i=1}^t z_i^{x_i}$, $r = \prod_{i=1}^t r_i^{x_i}$ and $u = \prod_{i=1}^t u_i^{x_i}$.
2. Using $\mathsf{VK} = \mathsf{VK}[1] \ldots \mathsf{VK}[L] \in \{0,1\}^L$, define the vector $\mathbf{f}_{\mathsf{VK}} = \mathbf{f}_{3,0} \cdot \prod_{i=1}^L \mathbf{f}_{3,i}^{\mathsf{VK}[i]}$ and assemble a Groth-Sahai CRS $\mathbf{f}_{\mathsf{VK}} = (\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_{\mathsf{VK}})$. Using $\mathbf{f}_{\mathsf{VK}}$, generate commitments $\mathbf{C}_z$, $\mathbf{C}_r$, $\mathbf{C}_u$ to the components of $(z, r, u) \in \mathbb{G}^3$ along with NIWI proofs $(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$ that $\mathbf{v}$ and (z, r, u) satisfy (3.1). Let $(\mathbf{C}_z, \mathbf{C}_r, \mathbf{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2) \in \mathbb{G}^{15}$ be the resulting commitments and proofs.
3. Generate $\sigma = \mathcal{S}(\mathsf{SK}, (\mathbf{v}, \mathbf{C}_z, \mathbf{C}_r, \mathbf{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \mathsf{lbl}))$ and output

$$\pi = (\mathsf{VK}, \mathbf{C}_z, \mathbf{C}_r, \mathbf{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \sigma) \tag{3.8}$$

$\mathsf{V}(\Gamma, \psi, \mathbf{v}, \pi, \mathsf{lbl})$**:** parse π as per (3.8) and $\mathbf{v}$ as $(v_1, \ldots, v_n) \in \mathbb{G}^n$. Return 1 if and only if

(i) $\mathcal{V}(\mathsf{VK}, (\mathbf{v}, \mathbf{C}_z, \mathbf{C}_r, \mathbf{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \mathsf{lbl}), \sigma) = 1$;

(ii) $(\mathbf{C}_z, \mathbf{C}_r, \mathbf{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$ forms a valid NIWI proof for the CRS $\mathbf{f}_{\mathsf{VK}} = (\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_{\mathsf{VK}})$, so that $\boldsymbol{\pi}_1 = (\pi_{1,1}, \pi_{1,2}, \pi_{1,3})$ and $\boldsymbol{\pi}_2 = (\pi_{2,1}, \pi_{2,2}, \pi_{2,3})$ satisfy

$$\begin{aligned}
\prod_{i=1}^n E\big(g_i, (1_{\mathbb{G}}, 1_{\mathbb{G}}, v_i)\big)^{-1} &= E\big(g_z, \mathbf{C}_z\big) \cdot E\big(g_r, \mathbf{C}_r\big) \cdot E(\pi_{1,1}, \mathbf{f}_1) \cdot E(\pi_{1,2}, \mathbf{f}_2) \cdot E(\pi_{1,3}, \mathbf{f}_{\mathsf{VK}}) \\
\prod_{i=1}^n E\big(h_i, (1_{\mathbb{G}}, 1_{\mathbb{G}}, v_i)\big)^{-1} &= E\big(h_z, \mathbf{C}_z\big) \cdot E\big(h, \mathbf{C}_u\big) \cdot E(\pi_{2,1}, \mathbf{f}_1) \cdot E(\pi_{2,2}, \mathbf{f}_2) \cdot E(\pi_{2,3}, \mathbf{f}_{\mathsf{VK}}).
\end{aligned}$$

To simulate a proof for a given vector $\mathbf{v} \in \mathbb{G}^n$, the simulator uses $\tau_{sim} = \mathsf{sk}_{hsps}$ to generate a fresh one-time homomorphic signature on $\mathbf{v} \in \mathbb{G}^n$ and proceeds as in steps 2-3 of P.

The proof π only consists of 15 group elements and a one-time pair (VK, σ). Remarkably, its length does not depend on the number of equations n or the number of variables t. In comparison, Groth-Sahai proofs already require $3t + 2n$ group elements in their basic form and become even more expensive when it comes to achieve unbounded simulation-soundness. The Jutla-Roy techniques [151] reduce the proof length to $2(n - t)$ elements – which only competes with our proofs when $t \approx n$ – but it is unclear how to extend them to get unbounded simulation-soundness without affecting their efficiency. Our CRS consists of $O(t + n + L)$ group elements against $O(t(n - t))$ in [151]. More detailed comparisons are given in Section 3.4.3 between proof systems based on the DLIN assumption.

Interestingly, the above scheme even outperforms Fiat-Shamir-like proofs derived from Σ-protocols which would give $\Theta(t)$-size proofs here. The construction readily extends to rely on the k-linear assumption for $k > 2$. In this case, the proof comprises $(k + 1)(2k + 1)$ elements and its size thus only depends on k, as detailed in the full version of [178].

Moreover, the verification algorithm only involves *linear* pairing product equations whereas all known unbounded simulation-sound extensions of Groth-Sahai proofs require either quadratic equations or a linearization step involving extra variables.

We finally remark that, if we give up the simulation-soundness property, the proof length drops to $k+1$ group elements under the k-linear assumption.

Theorem 7 ([178]). *The scheme is an unbounded simulation-sound QA-NIZK proof system if the DLIN assumption holds in $\mathbb{G}$ and Σ is strongly unforgeable.*

The above construction is not tightly secure as the gap between the simulation-soundness adversary's advantage and the probability to break the DLIN assumption depends on the number of simulated proofs obtained by the adversary. For applications like tight CCA2 security [142], it would be interesting to modify the proof system to obtain tight security.

3.4.2 Construction with (Single-Theorem) Relative Soundness

In applications where single-theorem relatively sound NIZK proofs suffice, we can further improve the efficiency. Under the k-linear assumption, the proof length reduces from $O(k^2)$ elements to $O(k)$ elements. Under the DLIN assumption, each proof fits within 4 elements and only costs $2n+6$ pairings to verify. In comparison, the verifier needs $2(n-t)(t+2)$ pairing evaluations in [151].

As in [150], we achieve relative soundness using smooth projective hash functions [90]. To this end, we need to encode the matrix $\rho \in \mathbb{G}^{t\times n}$ as a $2t \times (2n+1)$ matrix.

$\mathbb{K}_0(\lambda)$**:** choose symmetric bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ with $g \overset{\$}{\leftarrow} \mathbb{G}$. Then, output $\Gamma = (\mathbb{G}, \mathbb{G}_T, g)$.

Again, the dimensions of $\mathbf{A} \in \mathbb{Z}_p^{t\times n}$ can be either fixed or part of $\mathcal{L}_\rho$, so that t, n can be given as input to the CRS generation algorithm $\mathbb{K}_1$.

$\mathbb{K}_1(\Gamma, \rho)$**:** parse Γ as $(\mathbb{G}, \mathbb{G}_T, g)$ and ρ as $\rho = \big(G_{ij}\big)_{1\leq i\leq t,\ 1\leq j\leq n} \in \mathbb{G}^{t\times n}$ and do the following.

1. Choose two n-vectors $\mathbf{d} = (d_1, \ldots, d_n) \overset{\$}{\leftarrow} \mathbb{Z}_p^n$ and $\mathbf{e} = (e_1, \ldots, e_n) \overset{\$}{\leftarrow} \mathbb{Z}_p^n$ in order to define $\mathbf{W} = (W_1, \ldots, W_t) = g^{\mathbf{A}\cdot\mathbf{d}^\top} \in \mathbb{G}^t$ and $\mathbf{Y} = (Y_1, \ldots, Y_t) = g^{\mathbf{A}\cdot\mathbf{e}^\top} \in \mathbb{G}^t$. These will be used to define a projective hash function.
2. Generate a key pair $(\mathsf{pk}_{ots}, \mathsf{sk}_{ots})$ for the one-time linearly homomorphic signature of Section 3.2.1 in order to sign vectors in $\mathbb{G}^{2n+1}$. Let the public key be
$$\mathsf{pk}_{ots} = \big((\mathbb{G}, \mathbb{G}_T), g_z, g_r, h_z, h_u, \{(g_i, h_i)\}_{i=1}^{2n+1}\big)$$
and let $\mathsf{sk}_{ots} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^{2n+1}$ be the corresponding private key.
3. Use sk_{ots} to generate one-time homomorphic signatures $\{(z_i, r_i, u_i)\}_{i=1}^{2t}$ on the vectors below, which are obtained from the rows of the matrix $\rho = \big(G_{i,j}\big)_{1\leq i\leq t,\ 1\leq j\leq n}$.
$$\begin{array}{rcll} \mathbf{H}_{2i-1} & = & (G_{i,1}, \ldots, G_{i,n}, Y_i,\ 1\ ,\ \ldots\ ,1\ ) \in \mathbb{G}^{2n+1} & \qquad i \in \{1, \ldots, t\} \\ \mathbf{H}_{2i} & = & (1\ ,\ \ldots\ ,1\ , W_i, G_{i,1}, \ldots, G_{i,n}) \in \mathbb{G}^{2n+1} & \end{array}$$
4. Choose a collision-resistant hash function $H : \{0,1\}^* \rightarrow \mathbb{Z}_p$.

5. The CRS ψ consists of a first part $\mathbf{CRS}_1$ that is only used by the prover and a second part $\mathbf{CRS}_2$ which is only used by the verifier. These are defined as

$$\mathbf{CRS}_1 = \Big(\rho,\ \mathsf{pk}_{ots},\ \mathbf{W},\ \mathbf{Y},\ \{(z_i, r_i, u_i)\}_{i=1}^{2t},\ H\Big), \qquad \mathbf{CRS}_2 = \Big(\mathsf{pk}_{ots},\ \mathbf{W},\ \mathbf{Y},\ H\Big).$$

The simulation trapdoor τ_{sim} is sk_{ots} and the private verification trapdoor consists of $\tau_v = \{\mathbf{d}, \mathbf{e}\}$.

$\mathsf{P}(\Gamma, \psi, \mathbf{v}, x, \mathsf{lbl})$**:** given a candidate vector $\mathbf{v} \in \mathbb{G}^n$, a witness $\mathbf{x} = (x_1, \ldots, x_t) \in \mathbb{Z}_p^t$ such that $\mathbf{v} = g^{\mathbf{x}\cdot\mathbf{A}}$ and a label lbl, compute $\alpha = H(\rho, \mathbf{v}, \mathsf{lbl}) \in \mathbb{Z}_p$. Using $\{(z_i, r_i, u_i)\}_{i=1}^{2t}$, derive a one-time homomorphic signature (z, r, u) on $\tilde{\mathbf{v}} = \big(v_1, \ldots, v_n, \pi_0, v_1^\alpha, \ldots, v_n^\alpha\big) \in \mathbb{G}^{2n+1}$, where $\pi_0 = \prod_{i=1}^t (W_i^\alpha Y_i)^{x_i}$. Namely, compute and output $\pi = (z, r, u, \pi_0) \in \mathbb{G}^4$, where

$$z = \prod_{i=1}^t (z_{2i-1} \cdot z_{2i}^\alpha)^{x_i}, \quad r = \prod_{i=1}^t (r_{2i-1} \cdot r_{2i}^\alpha)^{x_i}, \quad u = \prod_{i=1}^t (u_{2i-1} \cdot u_{2i}^\alpha)^{x_i}, \quad \pi_0 = \prod_{i=1}^t (W_i^\alpha Y_i)^{x_i}$$

$\mathsf{V}(\Gamma, \psi, \mathbf{v}, \pi, \mathsf{lbl})$**:** parse the vector $\mathbf{v}$ as $(v_1, \ldots, v_n) \in \mathbb{G}^n$ and π as $(z, r, u, \pi_0) \in \mathbb{G}^4$. Compute $\alpha = H(\rho, \mathbf{v}, \mathsf{lbl})$ and return 1 if and only if the triple (z, r, u) is a valid signature on the vector $\tilde{\mathbf{v}} = (v_1, \ldots, v_n, \pi_0, v_1^\alpha, \ldots, v_n^\alpha) \in \mathbb{G}^{2n+1}$. Namely, it should satisfy the equalities

$$\begin{aligned} 1_{\mathbb{G}_T} &= e(g_z, z) \cdot e(g_r, r) \cdot \prod_{i=1}^n e(g_i \cdot g_{i+n+1}^\alpha, v_i) \cdot e(g_{n+1}, \pi_0) \qquad (3.9) \\ 1_{\mathbb{G}_T} &= e(h_z, z) \cdot e(h_u, u) \cdot \prod_{i=1}^n e(h_i \cdot h_{i+n+1}^\alpha, v_i) \cdot e(h_{n+1}, \pi_0). \end{aligned}$$

$\mathsf{W}(\Gamma, \psi, \tau_v, \mathbf{v}, \pi, \mathsf{lbl})$**:** given a vector $\mathbf{v} = (v_1, \ldots, v_n) \in \mathbb{G}^n$, parse π as $(z, r, u, \pi_0) \in \mathbb{G}^4$ and τ_v as $\{\mathbf{d}, \mathbf{e}\}$, with $\mathbf{d} = (d_1, \ldots, d_n) \in \mathbb{Z}_p^n$ and $\mathbf{e} = (e_1, \ldots, e_n) \in \mathbb{Z}_p^n$. Compute $\alpha = H(\rho, \mathbf{v}, \mathsf{lbl}) \in \mathbb{Z}_p$ and return 0 if the public verification test V fails. Otherwise, return 1 if $\pi_0 = \prod_{j=1}^n v_j^{e_j + \alpha d_j}$ and 0 otherwise.

We note that, while the proving algorithm is deterministic, each statement has many valid proofs. However, finding two valid proofs for the same statement is computationally hard, as we proved in [178].

The scheme readily extends to rest on the k-linear assumption with $k > 2$. In this case, the proof requires $k+2$ group elements – whereas combining the techniques of [150, 151] demands $k(n+1-t)$ elements per proof – and a CRS of size $O(k(n+t))$. Subsequently to our work [178], Jutla and Roy [153] and Abdalla, Ben Hamouda and Pointcheval [1] gave different constructions of one-time relatively-sound or simulation-sound QA-NIZK proofs made of only 3 group elements under the DLIN assumption.

Theorem 8 ([178])**.** *The above proof system is a relatively sound QA-NIZK proof system if the SDP assumption holds in* $(\mathbb{G}, \mathbb{G}_T)$ *and if* H *is a collision-resistant hash function.*

As an application, we showed in the full version of [178] how the DLIN-based version [235] of the Cramer-Shoup cryptosystem [88, 90] can be made publicly verifiable (meaning that well-formed ciphertext are recognizable given only the public key) by introducing only three group elements in the ciphertext. In the threshold setting, the resulting system can be distributed – without interaction during the decryption process – and proved secure against adaptive corruptions. As a result, we obtained [178] a new adaptively secure CCA2-secure *non-interactive* threshold cryptosystem based on the DLIN assumption with ciphertexts comprised of only 8 group elements. In comparison with the best previous variants [150, 151] of Cramer-Shoup with publicly verifiable ciphertexts, we thus spare one group element per ciphertext. If we compare our construction [178, Appendix I] with the first adaptively secure non-interactive threshold version of Cramer-Shoup [189], we shorten ciphertexts by 60%. The recent results of Jutla and Roy [153] yield further optimizations, which allow for ciphertexts made of 7 group elements under the DLIN assumptions (and even 5 group elements under the SXDH assumption).

Under the k-linear assumption, the scheme provides ciphertexts that are $\Theta(k)$ group elements shorter than in previous such constructions.

3.4.3 Comparisons

This section compares the various NIZK proofs of linear subspace membership based on the DLIN assumption. Comparisons are given in terms of CRS size, proof size and the number of pairing evaluations for the verifier.

In the table, we consider our basic proof system (without any form of simulation-soundness, where each proof is a one-time linearly homomorphic signature (z, r, u)), its unbounded simulation-sound variant and the relatively simulation-sound variant of Section 3.4.2. We compare these with the original Groth-Sahai proofs, their most efficient unbounded simulation-sound extensions due to Camenisch *et al.* [62] and the Jutla-Roy techniques [151, 153] with and without relative soundness.

Table 3.1: Comparison between proof systems for linear subspaces

Proof systems	CRS size$^{\diamondsuit\ *}$	Proof length$^{\diamondsuit}$	# of pairings† at verification
Groth-Sahai [138]	6	$3t+2n$	$3n(t+3)$
Jutla-Roy [151]	$4t(n-t)+3$	$2(n-t)$	2(n-t)(t+2)
Jutla-Roy RSS [151] + [150]	$4t(n+1-t)+3$	$2(n+1-t)+1$	$2(n+1-t)(t+2)$
Groth-Sahai USS [62]	18	$6t+2n+52^{\ddagger}$	$O(tn)$
Our basic QA-NIZK proofs	$2n+3t+4$	3	$2n+4$
Our RSS QA-NIZK proofs	$4n+8t+6$	4	$2n+6$
Our USS QA-NIZK proofs	$2n+3t+3L+10$	$20^{\ddagger}$	$2n+30$
Jutla-Roy [153]	$O(t+n)$	2	$2n+4$
Jutla-Roy RSS [151] + [153]	$O(t+n)$	3	$2n+4$
Abdalla *et al.*, one-time SS [1]	$O(t+n)$	3	$2n+4$

n: number of equations; t: number of variables; L: length of a hashed one-time verification key

$\diamondsuit$ These sizes are measured in terms of number of group elements.
$*$ The description $\rho \in \mathbb{G}^{t\times n}$ of the language is not counted as being part of the CRS here.
† The table does not consider optimizations using randomized batch verification techniques here.
‡ We consider instantiations using Groth's one-time signature [133], where verification keys and signatures consist of 3 group elements and two elements of $\mathbb{Z}_p$, respectively.

As can be observed in the table, our constructions all yield constant-size arguments. Moreover, the number of pairing evaluations is always independent of the number of variables t, which substantially fastens the verification process when $t \approx n/2$. The last three rows of the table consider the results that were subsequent to ours, including the implications of the techniques of Jutla and Roy [153] who independently proposed a different construction of constant-size QA-NIZK proofs of linear subspace membership. While their construction of [153] does not provide simulation-soundness, it can be combined with earlier results [150] so as to obtain a (one-time) relatively sound proof system with only 3 group elements per proof. It is unclear how to extend it into an unbounded simulation-sound proof system and the same holds for the construction of [1].

We also note that randomized batch verification techniques can be used to drastically reduce the number of pairing computations. In our USS system, for example, the number of pairings drops to $n + 18$ if the two verification equations are processed together and further optimizations are possible.

Our common reference strings always fit within $O(t + n)$ group elements (with another $O(L)$ elements in the USS variant) and thus provide significant savings w.r.t. [151] when $t \approx n/2$.

3.5 Conclusion

We gave new and somewhat unexpected applications of structure-preserving signatures in the construction of non-malleable cryptographic primitives like non-interactive non-malleable commitments, simulation-sound QA-NIZK proofs and chosen-ciphertext-secure public-key encryption. Paradoxically, these applications were made possible by first rendering structure-preserving signatures homomorphic (and thus malleable).

Beyond their applications to non-malleability, our LHSPS primitive is powerful enough to provide very simple realizations of constant-size QA-NIZK proofs of linear subspace membership. In fact, it is not hard to see that any one-time LHSPS system can be generically used to build such a QA-NIZK proof system. Moreover, the specific algebraic properties of our constructions made it possible to tweak them so as to obtain unbounded simulation-soundness without sacrificing the constant proof size. Via the technique of Malkin *et al.* [196], it is actually possible to combine the Groth-Sahai NIZK proofs with any LHSPS systems so as to build an USS QA-NIZK argument of subspace membership: the QA-NIZK proof can consist of a NIZK proof of knowledge of a linearly homomorphic signature. However, due to the use of Groth-Sahai NIZK proofs for pairing product equations, the resulting QA-NIZK proofs would not necessarily be of constant size. The constant proof length of our construction stems from the specific structure of the scheme which, via suitable information theoretic arguments in the security proof, allows us to only require NIWI (rather than NIZK) proofs of knowledge for pairing product equations.

Our constant-size QA-NIZK arguments recently allowed us [175] to improve upon the results of Chen and Wee [86], who gave signature schemes with almost tight security – meaning that the security loss only depends on the security parameter and not on the number of signing queries made by the adversary – under the K-linear assumption. Under the DLIN assumption, our construction allows reducing the signature length from 8 to 6 group elements. Our signature scheme [175] crucially relies on the fact that the size of proofs does not depend of the dimension of the considered subspace. It can be generalized to use any QA-NIZK argument of linear subspace membership. Hence, if the improved Jutla-Roy construction [153] is plugged into the high-level construction of [175], the signature length reduces to 5 group elements under the DLIN assumption and 3 elements under the SXDH assumption. The QA-NIZK proofs of [153] thus provide our construction with as short signatures as those of Blazy, Kiltz and Pan [36] with the benefit of shorter private keys.

Finally, together with Marc Joye and Moti Yung [174], we used our LHSPS systems to design (albeit in a non-generic manner) fully distributed non-interactive adaptively secure threshold signatures with round-optimal key generation. We expect our LHSPS primitive to find other applications in the future. For example, Catalano, Marcedone and Puglisi [79] recently used them to devise linearly homomorphic signatures which can operate in on-line/offline mode [106], by allowing expensive public-key operations to take place before the data to be signed is available.

Conclusion and Perspectives

Summary of Results

This manuscript highlighted the importance of structure-preserving cryptographic primitives and pairing-based non-interactive proof systems. Several applications were described with a focus on privacy-enhancing cryptographic techniques, like group encryption and group signatures, and non-malleable non-interactive primitives which include non-malleable commitments, simulation-sound QA-NIZK arguments of linear subspace membership and CCA2-secure encryption schemes.

Our contributions in the context of anonymity-related cryptography included the first efficient realization of the structure-preserving signature primitive suggested for the first time by Groth [133] in 2006. As an application of the more efficient SPS schemes proposed by Abe *et al.* [6, 4], we gave a novel and efficient solution to the venerable problem of conveniently revoking users in group signatures. Our most efficient revocable group signature [179] suitably combines structure-preserving signatures with other ingredients like the NNL Subset Cover [205] framework for broadcast encryption and the concise vector commitment scheme proposed by Moti Yung and myself in 2010 [188].

Surprisingly, the applications of structure-preserving signatures to non-malleability were made possible by first tweaking certain existing SPS schemes [6] so as to obtain linearly homomorphic (and thus malleable) structure-preserving signatures. Our construction of non-interactive non-malleable commitment to group elements is completely generic and can be based on any LHSPS realization. In their basic version (i.e., without the simulation-soundness property), our QA-NIZK arguments can also generically rely on any LHSPS scheme. In order to achieve unbounded simulation-soundness, our construction is no longer generic since its security proof relies on information-theoretic arguments which are specific to our concrete homomorphic LHSPS system.

Our results showed that structure-preserving signatures with homomorphic properties are a powerful primitive with unexpected applications. In a recent result [175], we also used them to design a more efficient variant of the Chen-Wee [86] signatures with a nearly tight security proof under the DLIN assumption (a similar result was independently obtained by Blazy *et al.* [36]). By applying techniques suggested in [192, 133, 3], we also obtained a more efficient construction of CCA2-secure public-key encryption scheme in the multi-challenge, multi-user setting[5] [30, 142]. In comparison with the best known construction with tight

[5] As shown in [30], the multi-user, multi-challenge CCA2 security of a cryptosystem is implied by its security in the single-user, single-challenge setting. However, the reduction is linearly affected by the number of users and the number of challenge ciphertexts per user. Tight multi-user, multi-challenge CCA2 security is thus generally non-trivial to prove.

multi-challenge CCA2 security [3], our technique reduces the ciphertext length from 398 to 69 group elements under the DLIN assumption. Together with Marc Joye and Moti Yung [174], we further used our specific one-time LHSPS scheme of Section 3.2.1 to build fully distributed non-interactive adaptively secure threshold signatures. We provide two optimally-resilient constructions – namely, one in the random oracle model and a slightly less efficient one in the standard model – with a one-round distributed key generation protocol in the erasure-free setting (meaning that the servers are not assumed to reliably erase all intermediate computation results in order to ensure security). To our knowledge, our constructions are the first non-interactive adaptively secure threshold signatures to simultaneously feature all these useful properties.

Directions for Future Work

Attribute-Based Encryption from QA-NIZK Proofs

We believe that other applications of linearly homomorphic structure-preserving signatures have not been explored yet. For example, they allowed us devise an ordinary digital signature scheme with a nearly tight reduction from a simple assumption in the standard model [175]. While, at first glance, this signature scheme appears amenable to constructing an identity-based encryption system (via the standard technique, notably used in [36], of randomizing the verification algorithm), we did not manage to formally prove it. In fact, while Jutla and Roy managed to construct a fully secure IBE system from their QA-NIZK arguments [151, Appendix H] via the dual system paradigm [249], we have not been able to build an IBE from our LHSPS schemes yet. One of my future objectives will be to fill this gap and further extend the realm of applications of the LHSPS primitive.

More generally, it will be interesting to determine the exact extent to which QA-NIZK proofs can be used to implement the dual system encryption paradigm [249, 171]. Jutla and Roy [151] used them in a non-generic way to build a very efficient IBE scheme with full security (as opposed to selective security [40]) under the SXDH assumption in prime order groups. Related results were obtained by Blazy *et al.* [36] via a more generic approach. However, both articles [151, 36] focus on the (hierarchical) IBE setting and it is unclear how to apply their techniques to get full security in attribute-based encryption [232, 132]. One of my upcoming goals will be to obtain a framework for building fully secure[6] attribute-based encryption schemes (in prime order groups) from QA-NIZK proofs by extending the dual system encryption method [249] in the same way as in [169, 215]. Ideally, the new framework should use QA-NIZK proofs so as to translate the techniques of Attrapadung [19] from composite order groups to prime order groups. This should notably provide us with fully secure unbounded attribute-based encryption systems for large universes [172, 228] and online/offline efficiency in prime order groups. Finally, extensions of the framework will be considered in order to use QA-NIZK proofs so as to build attribute-hiding functional encryption schemes (like inner product encryption [154]. In summary, my hope is to use QA-NIZK proofs in order to improve upon existing frameworks [170, 215, 87] for building fully secure IBE and related primitives [50] in prime order groups.

[6] Full security, as opposed to selective security [40], refers to the strongest security notion where the aversary can choose the attribute set of the challenge ciphertext in the challenge phase.

Better Constructions of Functional Encryption from Different Assumptions

In recent years, a renewed attention has been paid to lattice-based cryptography. Breakthrough results [122] showed how to safely implement efficient lattice-based signatures and identity-based encryption. It is even possible [74, 9] to construct hierarchical identity-based encryption (HIBE) schemes [123]. Despite certain improvements [10], currently available lattice-based HIBE schemes still have ciphertexts and private keys whose lengths depend on the depth of the hierarchy. The reason is that the latter always affects the dimension of underlying lattices in a way or another. In contrast, the world of bilinear maps allows HIBE schemes [43] with ciphertexts of constant size: their length only depends on the security parameter and not on the number of levels in the hierarchy or the depth of the receiver.

In the setting of an ongoing project on functional encryption, I am planning to investigate whether the aforementioned overhead is inherent to lattice-based cryptography. Should the answer be negative, I hope for a lattice-based analogue of [43] and aim at designing HIBE schemes with constant-size ciphertexts. This achievement would notably imply lattice-based forward-secure public-key encryption schemes with ciphertexts of constant (i.e, independent of the number of time periods) size and also open the way to lattice-based broadcast encryption with short ciphertexts. This would solve yet another challenging open problem as, for the time being, all broadcast encryption systems with short ciphertexts and private keys rely on ad hoc assumptions. In particular, we do not have a realization based on the standard learning-with-errors (LWE) assumption [227], let alone with adaptive security [124].

Another limitation of all known adaptively-secure lattice-based HIBE schemes [74, 9, 10] is that hierarchies are restricted to have a constant and small number of levels: indeed, a polynomial number of levels would translate into a non-polynomial reduction (and thus fail to provide any security guarantee) as the security bound exponentially declines with the number of levels. In order to sidestep the latter limitation, I thus hope to adapt suitable techniques from pairing-based cryptography [249] in the setting of lattices and obtain HIBE schemes supporting a polynomial number of levels with a polynomial reduction in their security proof. Ideally, I would like to obtain a fully secure lattice-based HIBE scheme (in the standard model) where the number of levels in the hierarchy does not need to be fixed when the system is set up. While such HIBE systems exist under discrete-logarithm-related assumptions [172], they remain elusive in the lattice world so far. It would also be interesting to extend those results so as to obtain full security in generalizations of (H)IBE such as attribute-based and functional encryption [50]. For the time being, we do not have a fully secure attribute-based encryption scheme based on standard lattice assumptions.

Efficient QA-NIZK Proofs for Lattice Problems

The quasi-adaptive setting [151] made it possible to improve upon the efficiency of existing NIZK proof systems in the standard model [151, 153, 178] for the specific language of linear subspaces in vector spaces spanned by vectors of group elements. An interesting open question is whether QA-NIZK proofs can be more efficiently obtained than regular NIZK proofs for specific problems involving lattices.

For example, given a random matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ defined over a prime modulus q and where $m = O(n \log q)$, it would be interesting to have QA-NIZK proofs for the LWE language $\mathcal{L} = \{\mathbf{v} \in \mathbb{Z}_q^m \mid \mathbf{v} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e},\ \mathbf{s} \in \mathbb{Z}_q^n\}$, where $\mathbf{e} \in \mathbb{Z}^m$ is a small-norm noise vector. This problem can be seen as a "subspace closeness" problem rather than a subspace membership problem: instead of putting the entries of $\mathbf{A}$ and $\mathbf{v}$ in the exponent, one adds a noise to the entries of $\mathbf{v}$. Unfortunately, our techniques of building QA-NIZK proofs from homomorphic signatures (described in Section 3.4) do not seem to carry over here. In particular, it seems difficult to apply them to the Boneh-Freeman linearly homomorphic signatures [47, 46]. The main difficulty is seemingly to guarantee the NIZK property while handling vectors of integers rather than vectors of group elements.

Solving this problem would help fill important gaps in lattice-based cryptography since, even in the random oracle model, efficient non-interactive zero-knowledge proof systems are only available for specific languages [200, 194, 146, 193, 141] so far. In the standard model, the best constructions we are aware of are those of Peikert and Vaikuntanathan [218], which are not known to apply to the LWE language. In the future, I am thus hoping to take steps towards filling this gap.

Bibliography

[1] M. Abdalla, F. Ben Hamouda, and D. Pointcheval. Disjunctions for hash proof systems: New constructions and applications. *Cryptology ePrint Archive: Report 2014/483*, 2014.

[2] M. Abe, M. Chase, B. David, M. Kohlweiss, R. Nishimaki, and M. Ohkubo. Constant-size structure-preserving signatures: Generic constructions and simple assumptions. In Wang and Sako [247], pages 4–24.

[3] M. Abe, B. David, M. Kohlweiss, R. Nishimaki, and M. Ohkubo. Tagged one-time signatures: Tight security and optimal tag size. In Kurosawa and Hanaoka [165], pages 312–331.

[4] M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, and M Ohkubo. Structure-preserving signatures and commitments to group elements. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010*, volume 6223 of *LNCS*, pages 209–236. Springer, 2010.

[5] M. Abe, J. Groth, K. Haralambiev, and M. Ohkubo. Optimal structure-preserving signatures in asymmetric bilinear groups. In Phillip Rogaway, editor, *CRYPTO*, volume 6841 of *Lecture Notes in Computer Science*, pages 649–666. Springer, 2011.

[6] M. Abe, K. Haralambiev, and M. Ohkubo. Signing on elements in bilinear groups for modular protocol design. *IACR Cryptology ePrint Archive*, 2010:133, 2010.

[7] M. Abe, K. Haralambiev, and M. Ohkubo. Group to group commitments do not shrink. In Pointcheval and Johansson [220], pages 301–317.

[8] T. Acar and L. Nguyen. Revocation for delegatable anonymous credentials. In Catalano et al. [80], pages 423–440.

[9] S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT*, pages 553–572, 2010.

[10] S. Agrawal, D. Boneh, and X. Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In *CRYPTO*, 2010.

[11] J. Ahn, D. Boneh, J. Camenisch, S. Hohenberger, A. Shelat, and B. Waters. Computing on authenticated data. In Cramer [92], pages 1–20.

[12] J. An, Y. Dodis, and T. Rabin. On the security of join signature and encryption. In Knudsen [161], pages 83–107.

[13] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song. Provable data possession at untrusted stores. In Ning et al. [212], pages 598–609.

[14] G. Ateniese, J. Camenisch, S. Hohenberger, and B. de Medeiros. Practical group signatures without random oracles. *IACR Cryptology ePrint Archive*, 2005:385, 2005.

[15] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *CRYPTO'00*, pages 255–270, 2000.

[16] G. Ateniese, S. Kamara, and J. Katz. Proofs of storage from homomorphic identification protocols. In Matsui [197], pages 319–333.

[17] G. Ateniese, D. Song, and G. Tsudik. Quasi-efficient revocation in group signatures. In *Financial Cryptography*, pages 183–197, 2002.

[18] Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick Drew McDaniel, editors. *Proceedings of the 11th ACM Conference on Computer and Communications Security, CCS 2004, Washington, DC, USA, October 25-29, 2004*. ACM, 2004.

[19] N. Attrapadung. Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In Nguyen and Oswald [211], pages 557–577.

[20] N. Attrapadung, K. Emura, G. Hanaoka, and Y. Sakai. A revocable group signature scheme from identity-based revocation techniques: Achieving constant-size revocation list. In *Applied Cryptography and Network Security (ACNS'14)*, pages 419–437, 2014.

[21] N. Attrapadung, F. Laguillaumie, J. Herranz, B. Libert, E. de Panafieu, and C. Ràfols. Attribute-based encryption schemes with constant-size ciphertexts. *Theoretical Computer Science*, (422):15–38, 2012.

[22] N. Attrapadung and B. Libert. Functional encryption for inner product: Achieving constant-size ciphertexts with adaptive security or support for negation. In *Public Key Cryptography*, pages 384–402, 2010.

[23] N. Attrapadung and B. Libert. Homomorphic network coding signatures in the standard model. In Catalano et al. [80], pages 17–34.

[24] N. Attrapadung, B. Libert, and E. de Panafieu. Expressive key policy attribute-based encryption with constant-size ciphertexts. In *Public Key Cryptography*, pages 90–108, 2011.

[25] N. Attrapadung, B. Libert, and T. Peters. Computing on authenticated data: New privacy definitions and constructions. In Wang and Sako [247], pages 367–385.

[26] N. Attrapadung, B. Libert, and T. Peters. Efficient completely context-hiding quotable and linearly homomorphic signatures. In Kurosawa and Hanaoka [165], pages 386–404.

[27] P. Barreto, B. Libert, N. McCullagh, and J.-J. Quisquater. Efficient and provably secure identity-based signatures and signcryption from bilinear maps. In *ASIACRYPT*, pages 515–532, 2005.

[28] M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya, and H. Shacham. Randomizable proofs and delegatable anonymous credentials. In Halevi [139], pages 108–125.

[29] M. Belenkiy, M. Chase, M. Kohlweiss, and A. Lysyanskaya. P-signatures and noninteractive anonymous credentials. In Ran Canetti, editor, *TCC*, volume 4948 of *Lecture Notes in Computer Science*, pages 356–374. Springer, 2008.

[30] M. Bellare, A. Boldyreva, and S. Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In *EUROCRYPT*, pages 259–274, 2000.

[31] M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 614–629. Springer, 2003.

[32] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *ACM Conference on Computer and Communications Security*, pages 62–73. ACM, 1993.

[33] M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. In *CT-RSA 2005*, volume 3376 of *LNCS*, pages 136–153. Springer, 2005.

[34] J. Benaloh and M. de Mare. One-way accumulators: A decentralized alternative to digital sinatures (extended abstract). In *EUROCRYPT*, pages 274–285, 1993.

[35] M. Blaze, G. Bleumer, and M. Strauss. Divertible protocols and atomic proxy cryptography. In *EUROCRYPT*, pages 127–144, 1998.

[36] O. Blazy, E. Kiltz, and J. Pan. (hierarchical) identity-based encryption from affine message authentication. In *CRYPTO*, 2014.

[37] M. Blum, A. de Santis, S. Micali, and G. Persiano. Noninteractive zero-knowledge. *SIAM J. Comput.*, 20(6):1084–1118, 1991.

[38] M. Blum, P. Feldman, and S. Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *STOC*, pages 103–112. ACM, 1988.

[39] A. Boldyreva, M. Fischlin, A. Palacio, and B. Warinschi. A closer look at PKI: Security and efficiency. In Catalano et al. [80], pages 458–475.

[40] D. Boneh and X. Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In Cachin and Camenisch [60], pages 223–238.

[41] D. Boneh and X. Boyen. Short signatures without random oracles. In Cachin and Camenisch [60], pages 56–73.

[42] D. Boneh and X. Boyen. Short signatures without random oracles and the SDH assumption in bilinear groups. *J. Cryptology*, 21(2):149–177, 2008.

[43] D. Boneh, X. Boyen, and E. Goh. Hierarchical identity based encryption with constant size ciphertext. In Cramer [91], pages 440–456.

[44] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In Franklin [110], pages 41–55.

[45] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In Kilian [159], pages 213–229.

[46] D. Boneh and D. Freeman. Homomorphic signatures for polynomial functions. In Paterson [217], pages 149–168.

[47] D. Boneh and D. Freeman. Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures. In Catalano et al. [80], pages 1–16.

[48] D. Boneh, D. Freeman, J. Katz, and B. Waters. Signing a linear subspace: Signature schemes for network coding. In Jarecki and Tsudik [147], pages 68–87.

[49] D. Boneh, C. Gentry, and B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In Victor Shoup, editor, *CRYPTO*, volume 3621 of *Lecture Notes in Computer Science*, pages 258–275. Springer, 2005.

[50] D. Boneh, A. Sahai, and B. Waters. Functional encryption: Definitions and challenges. In *TCC*, pages 253–273, 2011.

[51] D. Boneh, G. Segev, and B. Waters. Targeted malleability: homomorphic encryption for restricted computations. In Shafi Goldwasser, editor, *ITCS*, pages 350–366. ACM, 2012.

[52] D. Boneh and H. Shacham. Group signatures with verifier-local revocation. In Atluri et al. [18], pages 168–177.

[53] Dan Boneh, editor. *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*. Springer, 2003.

[54] X. Boyen and C. Delerablée. Expressive subgroup signatures. In *SCN*, pages 185–200, 2008.

[55] X. Boyen and B. Waters. Compact group signatures without random oracles. In Vaudenay [244], pages 427–444.

[56] X. Boyen and B. Waters. Full-domain subgroup hiding and constant-size group signatures. In *PKC 2007*, volume 4450 of *LNCS*, pages 1–15. Springer, 2007.

[57] E. Bresson and J. Stern. Efficient revocation in group signatures. In *Public Key Cryptography*, pages 190–206, 2001.

[58] E. Brickell. An efficient protocol for anonymously providing assurance of the container of the private key. In *Submission to the Trusted Computing Group*, 2003.

[59] E. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. In Atluri et al. [18], pages 132–145.

[60] Christian Cachin and Jan Camenisch, editors. *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, volume 3027 of *Lecture Notes in Computer Science*. Springer, 2004.

[61] J. Camenisch, R. Chaabouni, and A. shelat. Efficient protocols for set membership and range proofs. In Pieprzyk [219], pages 234–252.

[62] J. Camenisch, N. Chandran, and V. Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In Joux [149], pages 351–368.

[63] J. Camenisch, M. Dubovitskaya, and K. Haralambiev. Efficient structure-preserving signature scheme from standard assumptions. In *SCN*, pages 76–94, 2012.

[64] J. Camenisch, T. Groß, and T. Heydt-Benjamin. Rethinking accountable privacy supporting services: extended abstract. In *Digital Identity Management*, pages 1–8, 2008.

[65] J. Camenisch, K. Haralambiev, M. Kohlweiss, J. Lapon, and V. Naessens. Structure preserving CCA secure encryption and applications. In Lee and Wang [168], pages 89–106.

[66] J. Camenisch, M. Kohlweiss, and C. Soriente. An accumulator based on bilinear maps and efficient revocation for anonymous credentials. In Jarecki and Tsudik [147], pages 481–500.

[67] J. Camenisch, M. Kohlweiss, and C. Soriente. Solving revocation with efficient update of anonymous credentials. In *SCN*, pages 454–471, 2010.

[68] J. Camenisch and A. Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In Yung [250], pages 61–76.

[69] J. Camenisch, G. Neven, and M. Rückert. Fully anonymous attribute tokens from lattices. In *SCN*, pages 57–75, 2012.

[70] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *FOCS*, pages 136–145. IEEE Computer Society, 2001.

[71] R. Canetti and M. Fischlin. Universally composable commitments. In Kilian [159], pages 19–40.

[72] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited (preliminary version). In Vitter [245], pages 209–218.

[73] Ran Canetti and Juan A. Garay, editors. *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, volume 8043 of *Lecture Notes in Computer Science*. Springer, 2013.

[74] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In Gilbert [125].

[75] D. Catalano and D. Fiore. Vector commitments and their applications. In Kurosawa and Hanaoka [165], pages 55–72.

[76] D. Catalano, D. Fiore, and M. Messina. Zero-knowledge sets with short proofs. In Smart [239], pages 433–450.

[77] D. Catalano, D. Fiore, and B. Warinschi. Adaptive pseudo-free groups and applications. In Paterson [217], pages 207–223.

[78] D. Catalano, D. Fiore, and B. Warinschi. Efficient network coding signatures in the standard model. In Fischlin et al. [109], pages 680–696.

[79] D. Catalano, A Marcedone, and O. Puglisi. Authenticating computation on groups: New homomorphic primitives and applications. In *ASIACRYPT (2)*, pages 193–212, 2014.

[80] Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors. *Public Key Cryptography - PKC 2011 - 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, March 6-9, 2011. Proceedings*, volume 6571 of *Lecture Notes in Computer Science*. Springer, 2011.

[81] J. Cathalo, B. Libert, and M. Yung. Group encryption: Non-interactive realization in the standard model. In Matsui [197], pages 179–196.

[82] M. Chase and M. Kohlweiss. A new hash-and-sign approach and structure-preserving signatures from dlin. In *SCN*, pages 131–148, 2012.

[83] M. Chase, M. Kohlweiss, A. Lysyanskaya, and S. Meiklejohn. Malleable proof systems and applications. In Pointcheval and Johansson [220], pages 281–300.

[84] M. Chase, M. Kohlweiss, A. Lysyanskaya, and S. Meiklejohn. Verifiable elections that scale for free. In Pointcheval and Johansson [220], pages 479–496.

[85] D. Chaum and E. van Heyst. Group signatures. In *EUROCRYPT*, pages 257–265, 1991.

[86] J. Chen and H. Wee. Fully, (almost) tightly secure IBE from standard assumptions. In Canetti and Garay [73], pages 435–460.

[87] J. Chen and H. Wee. Dual system groups and its applications — compact HIBE and more. Cryptology ePrint Archive: Report 2014/265, April 2014.

[88] R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *CRYPTO*, volume 1462 of *Lecture Notes in Computer Science*, pages 13–25. Springer, 1998.

[89] R. Cramer and V. Shoup. Signature schemes based on the strong rsa assumption. In *ACM-CCS*, pages 46–51, 1999.

[90] R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Knudsen [161], pages 45–64.

[91] Ronald Cramer, editor. *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*. Springer, 2005.

[92] Ronald Cramer, editor. *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings*, volume 7194 of *Lecture Notes in Computer Science*. Springer, 2012.

[93] G. Di Crescenzo, Y. Ishai, and R. Ostrovsky. Non-interactive and non-malleable commitment. In Vitter [245], pages 141–150.

[94] I. Damgård and J. Groth. Non-interactive and reusable non-malleable commitment schemes. In Lawrence L. Larmore and Michel X. Goemans, editors, *STOC*, pages 426–437. ACM, 2003.

[95] C. Delerablée and D. Pointcheval. Dynamic fully anonymous short group signatures. In *VIETCRYPT*, pages 193–210, 2006.

[96] Y. Desmedt. Society and group oriented cryptography: A new concept. In Carl Pomerance, editor, *CRYPTO*, volume 293 of *Lecture Notes in Computer Science*, pages 120–127. Springer, 1987.

[97] Y. Desmedt. Computer security by redefining what a computer is. In *NSPW*, pages 160–166, 1993.

[98] Y. Desmedt and Y. Frankel. Threshold cryptosystems. In Gilles Brassard, editor, *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 307–315. Springer, 1989.

[99] Y. Dodis and N. Fazio. Public key broadcast encryption for stateless receivers. In Joan Feigenbaum, editor, *Digital Rights Management Workshop*, volume 2696 of *Lecture Notes in Computer Science*, pages 61–80. Springer, 2002.

[100] Y. Dodis, V. Shoup, and S. Walfish. Efficient constructions of composable commitments and zero-knowledge proofs. In Wagner [246], pages 515–535.

[101] D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography (extended abstract). In Cris Koutsougeras and Jeffrey Scott Vitter, editors, *STOC*, pages 542–552. ACM, 1991.

[102] D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.

[103] T. ElGamal. A public-key cryptosystem and a signature scheme based on discrete logarithms. In *CRYPTO*, pages 10–18, 1984.

[104] K. Emura, G. Hanaoka, G. Ohtake, T. Matsuda, and S. Yamada. Chosen ciphertext secure keyed-homomorphic public-key encryption. In Kurosawa and Hanaoka [165], pages 32–50.

[105] A. Escala, G. Herold, E. Kiltz, C. Ràfols, and J. L. Villar. An algebraic framework for diffie-hellman assumptions. In Canetti and Garay [73], pages 129–147.

[106] S. Even, O. Goldreich, and S. Micali. On-line/off-line digital schemes. In *CRYPTO*, pages 263–275, 1989.

[107] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.

[108] M. Fischlin, B. Libert, and M. Manulis. Non-interactive and re-usable universally composable string commitments with adaptive security. In Lee and Wang [168], pages 468–485.

[109] Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors. *Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings*, volume 7293 of *Lecture Notes in Computer Science*. Springer, 2012.

[110] Matthew K. Franklin, editor. *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*. Springer, 2004.

[111] D. Freeman. Improved security for linearly homomorphic signatures: A generic framework. In Fischlin et al. [109], pages 697–714.

[112] G. Fuchsbauer. Automorphic signatures in bilinear groups and an application to round-optimal blind signatures. *IACR Cryptology ePrint Archive*, 2009:320, 2009.

[113] G. Fuchsbauer and D. Pointcheval. Encrypting proofs on pairings and its application to anonymity for signatures. In *Pairing 2009*, pages 132–149, 2009.

[114] E. Fujisaki. New constructions of efficient simulation-sound commitments using encryption and their applications. In Orr Dunkelman, editor, *CT-RSA*, volume 7178 of *Lecture Notes in Computer Science*, pages 136–155. Springer, 2012.

[115] S. D. Galbraith, K. G. Paterson, and N. P. Smart. Pairings for cryptographers. *Discrete Appl. Math.*, 156(16), September 2008.

[116] J. Garay, P. MacKenzie, and K. Yang. Strengthening zero-knowledge protocols using signatures. In Eli Biham, editor, *EUROCRYPT*, volume 2656 of *Lecture Notes in Computer Science*, pages 177–194. Springer, 2003.

[117] Juan A. Garay, Atsuko Miyaji, and Akira Otsuka, editors. *Cryptology and Network Security, 8th International Conference, CANS 2009, Kanazawa, Japan, December 12-14, 2009. Proceedings*, volume 5888 of *Lecture Notes in Computer Science*. Springer, 2009.

[118] R. Gennaro. Multi-trapdoor commitments and their applications to proofs of knowledge secure under concurrent man-in-the-middle attacks. In Franklin [110], pages 220–236.

[119] R. Gennaro, C. Gentry, and B. Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In Rabin [225], pages 465–482.

[120] R. Gennaro, J. Katz, H. Krawczyk, and T. Rabin. Secure network coding over the integers. In Phong Q. Nguyen and David Pointcheval, editors, *Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 142–160. Springer, 2010.

[121] R. Gennaro and S. Micali. Independent zero-knowledge sets. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *ICALP (2)*, volume 4052 of *Lecture Notes in Computer Science*, pages 34–45. Springer, 2006.

[122] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206, 2008. Full version available at `http://eprint.iacr.org/2007/432.pdf`.

[123] C. Gentry and A. Silverberg. Hierarchical ID-based cryptography. In Yuliang Zheng, editor, *ASIACRYPT*, volume 2501 of *Lecture Notes in Computer Science*, pages 548–566. Springer, 2002.

[124] C. Gentry and B. Waters. Adaptive security in broadcast encryption systems (with short ciphertexts). In Joux [149], pages 171–188.

[125] Henri Gilbert, editor. *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*. Springer, 2010.

[126] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design. In *FOCS*, pages 174–187, 1986.

[127] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. In *STOC*, pages 291–304, 1985.

[128] S. Goldwasser and Y. Tauman. On the (in)security of the Fiat-Shamir paradigm. In *FOCS*, pages 102–113, 2003.

[129] S. Dov Gordon, J. Katz, and V . Vaikuntanathan. A group signature scheme from lattice assumptions. In Masayuki Abe, editor, *ASIACRYPT*, volume 6477 of *Lecture Notes in Computer Science*, pages 395–412. Springer, 2010.

[130] V. Goyal. Reducing trust in the PKG in identity-based cryptosystems. In *CRYPTO*, pages 430–447, 2007.

[131] V. Goyal, S. Lu, A. Sahai, and B. Waters. Black-box accountable authority identity-based encryption. In Ning et al. [213], pages 427–436.

[132] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ferng-Ching Lin, Der-Tsai Lee, Bao-Shuh Paul Lin, Shiuhpyng Shieh, and Sushil Jajodia, editors, *ACM Conference on Computer and Communications Security*, pages 195–203. ACM, 2006.

[133] J. Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In Xuejia Lai and Kefei Chen, editors, *ASIACRYPT*, volume 4284 of *Lecture Notes in Computer Science*, pages 444–459. Springer, 2006.

[134] J. Groth. Fully anonymous group signatures without random oracles. In Kurosawa [164], pages 164–180.

[135] J. Groth. Homomorphic trapdoor commitments to group elements. *IACR Cryptology ePrint Archive*, 2009:7, 2009.

[136] J. Groth, R. Ostrovsky, and A. Sahai. Non-interactive Zaps and new techniques for NIZK. In Vaudenay [244], pages 97–111.

[137] J. Groth, R. Ostrovsky, and A. Sahai. Perfect non-interactive zero knowledge for NP. In Vaudenay [244], pages 339–358.

[138] J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In Smart [239], pages 415–432.

[139] Shai Halevi, editor. *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*. Springer, 2009.

[140] D. Halevy and A. Shamir. The LSD broadcast encryption scheme. In Yung [250], pages 47–60.

[141] F. Ben Hamouda, J. Camenisch, S. Krenn, V. Lyubashevsky, and G. Neven. Better zero-knowledge proofs for lattice encryption and their application to group signatures. In *ASIACRYPT (1)*, pages 551–572, 2014.

[142] D. Hofheinz and T. Jager. Tightly secure signatures and public-key encryption. In Safavi-Naini and Canetti [230], pages 590–607.

[143] D. Hofheinz and E. Kiltz. Secure hybrid encryption from weakened key encapsulation. In *CRYPTO*, pages 553–571, 2007.

[144] J. Horwitz and B. Lynn. Toward hierarchical identity-based encryption. In Knudsen [161], pages 466–481.

[145] M. Izabachène, B. Libert, and D. Vergnaud. Block-wise p-signatures and non-interactive anonymous credentials with efficient attributes. In Liqun Chen, editor, *IMA Int. Conf.*, volume 7089 of *Lecture Notes in Computer Science*, pages 431–450. Springer, 2011.

[146] A. Jain, S. Krenn, K. Pietrzak, and A. Tentes. Commitments and efficient zero-knowledge proofs from learning parity with noise. In *ASIACRYPT*, pages 663–680, 2012.

[147] Stanislaw Jarecki and Gene Tsudik, editors. *Public Key Cryptography - PKC 2009, 12th International Conference on Practice and Theory in Public Key Cryptography, Irvine, CA, USA, March 18-20, 2009. Proceedings*, volume 5443 of *Lecture Notes in Computer Science*. Springer, 2009.

[148] R. Johnson, D. Molnar, D. Song, and D. Wagner. Homomorphic signature schemes. In Bart Preneel, editor, *CT-RSA*, volume 2271 of *Lecture Notes in Computer Science*, pages 244–262. Springer, 2002.

[149] Antoine Joux, editor. *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, volume 5479 of *Lecture Notes in Computer Science*. Springer, 2009.

[150] C. Jutla and A. Roy. Relatively-sound NIZKs and password-based key-exchange. In Fischlin et al. [109], pages 485–503.

[151] C. Jutla and A. Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT (1)*, volume 8269 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2013.

[152] C. Jutla and A. Roy. Dual-system simulation-soundness with applications to UC-PAKE and more. *Cryptology ePrint Archive: Report 2014/805*, 2014.

[153] C. Jutla and A. Roy. Switching lemma for bilinear tests and constant-size NIZK proofs for linear subspaces. In *CRYPTO (2)*, pages 295–312, 2014.

[154] J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Smart [239], pages 146–162.

[155] A. Kiayias, Y. Tsiounis, and M. Yung. Traceable signatures. In Cachin and Camenisch [60], pages 571–589.

[156] A. Kiayias, Y. Tsiounis, and M. Yung. Group encryption. In Kurosawa [164], pages 181–199.

[157] A. Kiayias and M. Yung. Group signatures with efficient concurrent join. In Cramer [91], pages 198–214.

[158] A. Kiayias and M. Yung. Secure scalable group signature with dynamic joins and separable authorities. *IJSN*, 1(1/2):24–45, 2006.

[159] Joe Kilian, editor. *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, volume 2139 of *Lecture Notes in Computer Science*. Springer, 2001.

[160] E. Kiltz. Chosen-ciphertext security from tag-based encryption. In *TCC'06*, volume 3876 of *LNCS*, pages 581–600. Springer, 2006.

[161] Lars R. Knudsen, editor. *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, volume 2332 of *Lecture Notes in Computer Science*. Springer, 2002.

[162] H. Krawczyk and T. Rabin. Chameleon signatures. In *NDSS*, 2000.

[163] S. Kunz-Jacques and D. Pointcheval. About the security of MTI/C0 and MQV. In Roberto De Prisco and Moti Yung, editors, *SCN*, volume 4116 of *Lecture Notes in Computer Science*, pages 156–172. Springer, 2006.

[164] Kaoru Kurosawa, editor. *Advances in Cryptology - ASIACRYPT 2007, 13th International Conference on the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, December 2-6, 2007, Proceedings*, volume 4833 of *Lecture Notes in Computer Science*. Springer, 2007.

[165] Kaoru Kurosawa and Goichiro Hanaoka, editors. *Public-Key Cryptography - PKC 2013 - 16th International Conference on Practice and Theory in Public-Key Cryptography, Nara, Japan, February 26 - March 1, 2013. Proceedings*, volume 7778 of *Lecture Notes in Computer Science*. Springer, 2013.

[166] F. Laguillaumie, A. Langlois, B. Libert, and D. Stehlé. Lattice-based group signatures with logarithmic signature size. In *ASIACRYPT*, pages 41–61, 2013.

[167] F. Laguillaumie, P. Paillier, and D. Vergnaud. Universally convertible directed signatures. In Roy [229], pages 682–701.

[168] Dong Hoon Lee and Xiaoyun Wang, editors. *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*. Springer, 2011.

[169] A. Lewko. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In Gilbert [125], pages 62–91.

[170] A. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In Pointcheval and Johansson [220], pages 318–335.

[171] A. Lewko and B. Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In Micciancio [201], pages 455–479.

[172] A. Lewko and B. Waters. Unbounded HIBE and attribute-based encryption. In Paterson [217], pages 547–567.

[173] B. Libert and M. Joye. Group signatures with message-dependent opening in the standard model. In *CT-RSA*, pages 286–306, 2014.

[174] B. Libert, M. Joye, and M. Yung. Born and raised distributed: Fully distributed non-interactive adaptively secure threshold signatures with short shares. In *PODC*, pages 303–312. ACM Press, 2014.

[175] B. Libert, M. Joye, M. Yung, and T. Peters. Concise multi-challenge CCA-secure encryption and signatures with almost tight security. In *ASIACRYPT (2)*, pages 1–21, 2014.

[176] B. Libert, M. Joye, M. Yung, and T. Peters. Traceable group encryption. In Canetti and Garay [73], pages 592–610.

[177] B. Libert, T. Peters, M. Joye, and M. Yung. Linearly homomorphic structure-preserving signatures and their applications. In Canetti and Garay [73], pages 289–307.

[178] B. Libert, T. Peters, M. Joye, and M. Yung. Non-malleability from malleability: Simulation-sound quasi-adaptive NIZK proofs and CCA2-secure encryption from homomorphic signatures. In Nguyen and Oswald [211].

[179] B. Libert, T. Peters, and M. Yung. Group signatures with almost-for-free revocation. In Safavi-Naini and Canetti [230], pages 571–589.

[180] B. Libert, T. Peters, and M. Yung. Scalable group signatures with revocation. In Pointcheval and Johansson [220], pages 609–627.

[181] B. Libert, J.-J. Quisquater, and M. Yung. Foward-secure signatures in untrusted update environments: Efficient and generic constructions. In Ning et al. [212], pages 511–520.

[182] B. Libert, J.-J. Quisquater, and M. Yung. Key evolution systems in untrusted update environments. *ACM Transactions on Information and Systems Security*, 13(4), 2010.

[183] B. Libert and D. Vergnaud. Multi-use unidirectional proxy re-signatures. In Ning et al. [213], pages 511–520.

[184] B. Libert and D. Vergnaud. Unidirectional chosen-ciphertext-secure proxy re-encryption. In Ronald Cramer, editor, *PKC*, volume 4939 of *Lecture Notes in Computer Science*, pages 360–379. Springer, 2008.

[185] B. Libert and D. Vergnaud. Group signatures with verifier-local revocation and backward unlinkability in the standard model. In Garay et al. [117], pages 498–517.

[186] B. Libert and D. Vergnaud. Towards black-box accountable authority IBE with short ciphertexts and private keys. In Garay et al. [117], pages 235–255.

[187] B. Libert and M. Yung. Efficient traceable signatures in the standard model. In *Pairing*, pages 187–205, 2009.

[188] B. Libert and M. Yung. Concise mercurial vector commitments and independent zero-knowledge sets with short proofs. In Micciancio [201], pages 499–517.

[189] B. Libert and M. Yung. Adaptively secure forward-secure non-interactive threshold cryptosystems. In Chuankun Wu, Moti Yung, and Dongdai Lin, editors, *Inscrypt*, volume 7537 of *Lecture Notes in Computer Science*, pages 1–21. Springer, 2011.

[190] B. Libert and M. Yung. Efficient traceable signatures in the standard model. *Theoretical Computer Science*, 412(12-14):1220–1242, 2011.

[191] B. Libert and M. Yung. Non-interactive CCA-secure threshold cryptosystems with adaptive security: New framework and constructions. In Cramer [92], pages 75–93.

[192] Y. Lindell. A simple construction of CCA2-secure public-key encryption under general assumptions. In Menezes [198], pages 241–254.

[193] S. Ling, K. Nguyen, D. Stehlé, and H. Wang. Improved zero-knowledge proofs of knowledge for the isis problem, and applications. In *Public Key Cryptography*, pages 107–124, 2013.

[194] V. Lyubashevsky. Lattice-based identification schemes secure under active attacks. In *PKC*, pages 162–179, 2014.

[195] P. MacKenzie and K. Yang. On simulation-sound trapdoor commitments. In Cachin and Camenisch [60], pages 382–400.

[196] T. Malkin, I. Teranishi, Y. Vahlis, and M. Yung. Signatures resilient to continual leakage on memory and computation. In Yuval Ishai, editor, *TCC*, volume 6597 of *Lecture Notes in Computer Science*, pages 89–106. Springer, 2011.

[197] Mitsuru Matsui, editor. *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*, volume 5912 of *Lecture Notes in Computer Science*. Springer, 2009.

[198] Alfred Menezes, editor. *Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings*, volume 4622 of *Lecture Notes in Computer Science*. Springer, 2007.

[199] S. Micali, M. Rabin, and J. Kilian. Zero-knowledge sets. In *FOCS*, pages 80–91, 2003.

[200] D. Micciancio and S. Vadhan. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In *CRYPTO*, pages 282–298, 2003.

[201] Daniele Micciancio, editor. *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings*, volume 5978 of *Lecture Notes in Computer Science*. Springer, 2010.

[202] T. Nakanishi, H. Fujii, Y. Hira, and N. Funabiki. Revocable group signature schemes with constant costs for signing and verifying. In Jarecki and Tsudik [147], pages 463–480.

[203] T. Nakanishi and N. Funabiki. Verifier-local revocation group signature schemes with backward unlinkability from bilinear maps. In Roy [229], pages 533–548.

[204] T. Nakanishi and N. Funabiki. Revocable group signatures with compact revocation list using accumulators. In *ICISC*, pages 435–451, 2013.

[205] D. Naor, M. Naor, and J. Lotspiech. Revocation and tracing schemes for stateless receivers. In Kilian [159], pages 41–62.

[206] M. Naor. On cryptographic assumptions and challenges. In Boneh [53], pages 96–109.

[207] M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *STOC*, pages 33–43, 1989.

[208] M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In Harriet Ortiz, editor, *STOC*, pages 427–437. ACM, 1990.

[209] L. Nguyen. Accumulators from bilinear pairings and applications. In Alfred Menezes, editor, *CT-RSA*, volume 3376 of *Lecture Notes in Computer Science*, pages 275–292. Springer, 2005.

[210] L. Nguyen and R. Safavi-Naini. Efficient and provably secure trapdoor-free group signature schemes from bilinear pairings. In Pil Joong Lee, editor, *ASIACRYPT*, volume 3329 of *Lecture Notes in Computer Science*, pages 372–386. Springer, 2004.

[211] Phong Q. Nguyen and Elisabeth Oswald, editors. *Advances in Cryptology - EUROCRYPT 2014, 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, Lecture Notes in Computer Science. Springer, 2014.

[212] Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors. *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, October 28-31, 2007*. ACM, 2007.

[213] Peng Ning, Paul F. Syverson, and Somesh Jha, editors. *Proceedings of the 2008 ACM Conference on Computer and Communications Security, CCS 2008, Alexandria, Virginia, USA, October 27-31, 2008*. ACM, 2008.

[214] R. Nishimaki, E. Fujisaki, and K. Tanaka. A multi-trapdoor commitment scheme from the RSA assumption. In *ACISP*, pages 182–199, 2010.

[215] T. Okamoto and K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In Rabin [225], pages 191–208.

[216] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT'99*, LNCS, pages 223–238. Springer, 1999.

[217] Kenneth G. Paterson, editor. *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, volume 6632 of *Lecture Notes in Computer Science*. Springer, 2011.

[218] C. Peikert and V. Vaikuntanathan. Noninteractive statistical zero-knowledge proofs for lattice problems. In *CRYPTO*, pages 536–553. Springer, 2008.

[219] Josef Pieprzyk, editor. *Advances in Cryptology - ASIACRYPT 2008, 14th International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, Australia, December 7-11, 2008. Proceedings*, volume 5350 of *Lecture Notes in Computer Science*. Springer, 2008.

[220] David Pointcheval and Thomas Johansson, editors. *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, volume 7237 of *Lecture Notes in Computer Science*. Springer, 2012.

[221] M. Prabhakaran and M. Rosulek. Rerandomizable RCCA encryption. In Menezes [198], pages 517–534.

[222] M. Prabhakaran and M. Rosulek. Homomorphic encryption with CCA security. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP (2)*, volume 5126 of *Lecture Notes in Computer Science*, pages 667–678. Springer, 2008.

[223] M. Prabhakaran and M. Rosulek. Towards robust computation on encrypted data. In Pieprzyk [219], pages 216–233.

[224] B. Qin, Q. Wu, W. Susilo, Y. Mu, and Y. Wang. Publicly verifiable privacy-preserving group decryption. In *Inscrypt*, LNCS, pages 72–83. Springer, 2008.

[225] Tal Rabin, editor. *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, volume 6223 of *Lecture Notes in Computer Science*. Springer, 2010.

[226] C. Rackoff and D. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *CRYPTO*, volume 576 of *Lecture Notes in Computer Science*, pages 433–444. Springer, 1991.

[227] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93, 2005.

[228] Y. Rouselakis and B. Waters. Practical constructions and new proof methods for large universe attribute-based encryption. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM Conference on Computer and Communications Security*, pages 463–474. ACM, 2013.

[229] Bimal K. Roy, editor. *Advances in Cryptology - ASIACRYPT 2005, 11th International Conference on the Theory and Application of Cryptology and Information Security, Chennai, India, December 4-8, 2005, Proceedings*, volume 3788 of *Lecture Notes in Computer Science*. Springer, 2005.

[230] Reihaneh Safavi-Naini and Ran Canetti, editors. *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, volume 7417 of *Lecture Notes in Computer Science*. Springer, 2012.

[231] A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *FOCS*, pages 543–553. IEEE Computer Society, 1999.

[232] A. Sahai and B. Waters. Fuzzy identity-based encryption. In Cramer [91], pages 457–473.

[233] Y. Sakai, K. Emura, G. Hanaoka, Y. Kawai, T. Matsuda, and K. Omote. Group signatures with message-dependent opening. In *Pairing*, pages 270–294, 2012.

[234] M. Scott. Authenticated ID-based key exchange and remote log-in with simple token and pin number. Technical report, Cryptology ePrint Archive: Report 2002/164, 2002.

[235] H. Shacham. A cramer-shoup encryption scheme from the linear assumption and from progressively weaker linear variants. *IACR Cryptology ePrint Archive*, page 74, 2007.

[236] A. Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages 47–53, 1984.

[237] V. Shoup. Lower bounds for discrete logarithms and related problems. In *EUROCRYPT*, pages 256–266, 1997.

[238] V. Shoup. A proposal for an ISO standard for public key encryption (version 2.1). Manuscript, December 2001.

[239] Nigel P. Smart, editor. *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, volume 4965 of *Lecture Notes in Computer Science*. Springer, 2008.

[240] D. Song. Practical forward secure group signature schemes. In *ACM Conference on Computer and Communications Security*, pages 225–234, 2001.

[241] P. Tsang, M. Ho Au, A. Kapadia, and S. Smith. Blacklistable anonymous credentials: blocking misbehaving users without ttps. In Ning et al. [212], pages 72–81.

[242] P. Tsang, M. Ho Au, A. Kapadia, and S. Smith. Perea: towards practical ttp-free revocation in anonymous authentication. In Ning et al. [213], pages 333–344.

[243] G. Tsudik and S. Xu. Accumulating composites and improved group signing. In Chi-Sung Laih, editor, *ASIACRYPT*, volume 2894 of *Lecture Notes in Computer Science*, pages 269–286. Springer, 2003.

[244] Serge Vaudenay, editor. *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*. Springer, 2006.

[245] Jeffrey Scott Vitter, editor. *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*. ACM, 1998.

[246] David Wagner, editor. *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*, volume 5157 of *Lecture Notes in Computer Science*. Springer, 2008.

[247] Xiaoyun Wang and Kazue Sako, editors. *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*. Springer, 2012.

[248] B. Waters. Efficient identity-based encryption without random oracles. In Cramer [91], pages 114–127.

[249] B. Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In Halevi [139], pages 619–636.

[250] Moti Yung, editor. *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, volume 2442 of *Lecture Notes in Computer Science*. Springer, 2002.

[251] S. Zhou and D. Lin. Shorter verifier-local revocation group signatures from bilinear maps. In *CANS*, pages 126–143, 2006.

Appendix

Group Encryption: Non-Interactive Realization in the Standard Model

Julien Cathalo[1] ⋆, Benoît Libert[1] ⋆⋆, and Moti Yung[2]

[1] Université catholique de Louvain, Crypto Group (Belgium)
[2] Google Inc. and Columbia University (USA)

Abstract. Group encryption (GE) schemes, introduced at Asiacrypt'07, are an encryption analogue of group signatures with a number of interesting applications. They allow a sender to encrypt a message (in the CCA2 security sense) for some member of a PKI group concealing that member's identity (in a CCA2 security sense, as well); the sender is able to convince a verifier that, among other things, the ciphertext is valid and some anonymous certified group member will be able to decrypt the message. As in group signatures, an opening authority has the power of pinning down the receiver's identity. The initial GE construction uses interactive proofs as part of the design (which can be made non-interactive using the random oracle model) and the design of a fully non-interactive group encryption system is still an open problem. In this paper, we give the first GE scheme, which is a pure encryption scheme in the standard model, *i.e.*, a scheme where the ciphertext is a single message and proofs are non-interactive (and do not employ the random oracle heuristic). As a building block, we use a new public key certification scheme which incurs the smallest amount of interaction, as well.

Keywords. Group encryption, anonymity, provable security.

1 Introduction

Group encryption (GE) schemes, introduced by Kiayias, Tsiounis and Yung [29], are the encryption analogue of group signatures [16]. The latter primitives basically allow a group member to sign messages in the name of a group without revealing his identity. In a similar spirit, GE systems aim to hide the identity of a ciphertext's recipient and still guarantee that he belongs to a population of registered members in a group administered by a group manager (GM). A sender can generate an anonymous encryption of some plaintext m intended for a receiver holding a public key that was certified by the GM (message security and receiver anonymity being both in the CCA2 sense). The ciphertext is prepared while leaving an opening authority (OA) the ability to "open" the ciphertext (analogously to the opening operation in group signatures) and uncover the receiver's name. At the same time, the sender should be able to convince a verifier that (1) the ciphertext is a valid encryption under the public key of some group member holding a valid certificate; (2) if necessary, the opening authority will be able to find out who the receiver is; (3) (optionally) the plaintext is a witness satisfying some public relation.

Motivations. The GE primitive was motivated by various privacy applications such as anonymous trusted third parties or oblivious retriever storage. Many cryptographic protocols such as fair exchange, fair encryption or escrow encryption, involve trusted third parties that remain offline most of the time and are only involved to resolve problems. Group encryption allows one to verifiably encrypt some message to such a trusted third party while hiding his identity among a set of possible

⋆ This author's research was supported by the Belgian Walloon Region project ALAWN (Programme Wist 2).

⋆⋆ This author acknowledges the Belgian National Fund for Scientific Research (F.R.S.-F.N.R.S.) for their financial support.

trustees. For instance, a user can encrypt a key (e.g., in an "international key escrow system") to his own national trusted representative without letting the ciphertext reveal the latter's identity, which could leak information on the user's citizenship. At the same time, everyone can be convinced that the ciphertext is heading for an authorized trustee.

Group encryption also finds applications in ubiquitous computing, where anonymous credentials must be transferred between peer devices belonging to the same group. Asynchronous transfers may require to involve an untrusted storage server to temporarily store encrypted credentials. In such a situation, GE schemes may be used to simultaneously guarantee that (1) the server retains properly encrypted valid credentials that it cannot read; (2) credentials have a legitimate anonymous retriever; (3) if necessary, an authority will be able to determine who the retriever is.

By combining cascaded group encryptions using multiple trustees and according to a sequence of identity discoveries and transfers, one can also implement group signatures where signers can flexibly specify how a set of trustees should operate to open their signatures.

Prior Works. Kiayias, Tsiounis and Yung (KTY) [29] formalized the concept of group encryption and provided a suitable security modeling. They presented a modular design of GE system and proved that, beyond zero-knowledge proofs, anonymous public key encryption schemes with CCA2 security, digital signatures, and equivocal commitments are necessary to realize the primitive. They also showed how to efficiently instantiate their general construction using Paillier's cryptosystem [35] (or, more precisely, a modification of the Camenisch-Shoup [13] variant of Paillier). While efficient, their scheme is not a single message encryption, since it requires the sender to interact with the verifier in a Σ-protocol to convince him that the aforementioned properties are satisfied. Interaction can be removed using the Fiat-Shamir paradigm [20] (and thus the random oracle model [4]), but only heuristic arguments [22] (see also [14]) are then possible in terms of security.

Independently, Qin *et al.* [36] considered a closely related primitive with non-interactive proofs and short ciphertexts. However, they avoid interaction by explicitly employing a random oracle and also rely on strong interactive assumptions. As we can see, none of these schemes is a truly non-interactive encryption scheme without the random oracle idealization.

Our Contribution. As already noted in various contexts such as anonymous credentials [2], rounds of interaction are expensive and even impossible at times as, in some applications, proofs should be verifiable by third parties that are not present when provers are available. In the setting of group encryption, this last concern is even more constraining as it requires the sender, who may be required to repeat proofs with many verifiers, to maintain a state and remember the random coins that he uses to encrypt *every* single ciphertext. In the frequent situation where many encryptions have to be generated using independent random coins, this becomes a definite bottleneck.

This paper solves the above problems and describes the first realization of group encryption which is a fully non-interactive encryption scheme with CCA2-security and anonymity in the standard model. In our scheme, senders do not need to maintain a state: thanks to the Groth-Sahai [27] non-interactive proof systems, the proof of a ciphertext can be generated once-and-for-all at the same time as the ciphertext itself. Furthermore, using suitable parameters and for a comparable security level, we can also shorten ciphertexts by a factor of 2 in comparison with the KTY scheme. As far as communication goes, the size of proofs allows decreasing by more than 75% the number of transmitted bits between the sender and the verifier.

Since our goal is to avoid interaction, we also design a joining protocol (*i.e.*, a protocol whereby the user effectively becomes a group member and gets his public key certified by the GM) which requires the smallest amount of interaction: as in the Kiayias-Yung group signature [30], only two messages have to be exchanged between the GM and the user and the latter need not to prove anything about his public key. In particular, rewinding is not necessary in security proofs and the join protocol can be safely executed in a concurrent environment, when many users want to register at the same time. The join protocol uses a non-interactive public key certification scheme where discrete-logarithm-type public keys can be signed as if they were ordinary messages (and without knowing the matching private key) while leaving the ability to efficiently prove knowledge of the certificate/public key using the Groth-Sahai techniques. To certify users without having to rewind[3] in security proofs, the KTY scheme uses groups of hidden order (and more precisely, Camenisch-Lysyanskaya signatures [12]). In public order groups, to the best of our knowledge, our construction is the first certification method that does not require any form of proof of knowledge of private keys. We believe it to be of independent interest as it can be used to construct group signatures (in the standard model) where the joining mechanism tolerates concurrency in the model of [30] without demanding more than two moves of interaction.

Organization. In section 2, we describe the intractability assumptions that we need and recall the KTY model of group encryption. Section 3 explains the building blocks of our construction and notably describes our certification scheme. Our GE system is depicted in section 4.

2 Background

In the paper, when S is a set, $x \overset{\$}{\leftarrow} S$ denotes the action of choosing x at random in S. By $a \in \mathsf{poly}(\lambda)$, we mean that a is a polynomial in λ while $b \in \mathsf{negl}(\lambda)$ says that b is a negligible function of λ. When a and b are two binary strings, $a||b$ stands for their concatenation.

2.1 Complexity Assumptions

We use groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order p with an efficiently computable map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ such that $e(g^a, h^b) = e(g, h)^{ab}$ for any $(g, h) \in \mathbb{G} \times \mathbb{G}$, $a, b \in \mathbb{Z}$ and $e(g, h) \neq 1_{\mathbb{G}_T}$ whenever $g, h \neq 1_{\mathbb{G}}$.

In this setting, we rely on an assumption introduced in [7] that allows constructing efficient non-interactive proofs as pointed out in [27].

Definition 1. *The* **Decision Linear Problem** *(DLIN) in $\mathbb{G}$, is to distinguish the distribution of linear tuples $D_1 = \{(g, g^a, g^b, g^{ac}, g^{bd}, g^{c+d}) | a, b, c, d \overset{\$}{\leftarrow} \mathbb{Z}_p^*\}$ from the distribution of random tuples $D_2 = \{(g, g^a, g^b, g^{ac}, g^{bd}, g^z) | a, b, c, d, z \overset{\$}{\leftarrow} \mathbb{Z}_p^*\}$. The* **Decision Linear Assumption** *is the intractability of DLIN for any PPT algorithm $\mathcal{D}$.*

This problem amounts to deciding whether vectors $\vec{g_1} = (g^a, 1, g)$, $\vec{g_2} = (1, g^b, g)$ and $\vec{g_3}$ are linearly dependent or not. We also consider a related computational problem which bears similarities with simultaneous pairing problems [26, 25].

Definition 2. *The* **Simultaneous Double Pairing problem** *(S2P) in $\mathbb{G}$ is, given a tuple of elements $(g_1, g_2, g_{1,c}, g_{2,d}) \in \mathbb{G}^4$, to find a non-trivial triple $(u, v, w) \in \mathbb{G}^3 \backslash \{(1_{\mathbb{G}}, 1_{\mathbb{G}}, 1_{\mathbb{G}})\}$ such that $e(g_1, u) = e(g_{1,c}, w)$ and $e(g_2, v) = e(g_{2,d}, w)$.*

[3] Although the simulator does not need to rewind proofs of knowledge in [29], users still have to interactively prove the validity of their public key.

Like the simultaneous triple pairing assumption [25], the hardness of this problem is implied by the DLIN assumption: given $(g, g_1, g_2, g_1^c, g_2^d, \eta \overset{?}{=} g^{c+d})$ any algorithm that, on input of (g_1, g_2, g_1^c, g_2^d), outputs a non-trivial (u, v, w) such that $e(g_1, u) = e(g_1^c, w)$, $e(g_2, v) = e(g_2^d, w)$ allows telling whether $\eta = g^{c+d}$ by testing if $e(g, u \cdot v) = e(\eta, w)$ (since $u = w^c$ and $v = w^d$).

We also use the Hidden Strong Diffie-Hellman (HSDH) assumption introduced in [10] as a strengthening of the Strong Diffie-Hellman assumption [6].

Definition 3. *The* **ℓ-Hidden Strong Diffie-Hellman problem** *(ℓ-HSDH) in $\mathbb{G}$ consists in, given $(g, \Omega = g^\omega, u) \xleftarrow{\$} \mathbb{G}^3$ and triples $(g^{1/(\omega+s_i)}, g^{c_i}, u^{c_i})$ with $c_1, \ldots, c_\ell \xleftarrow{\$} \mathbb{Z}_p^*$, finding another triple $(g^{1/(\omega+c)}, g^c, u^c)$ such that $c \neq c_i$ for $i = 1, \ldots, \ell$.*

We finally need the following variant of the Diffie-Hellman assumption.

Definition 4. *The* **Flexible Diffie-Hellman problem** *(FlexDH) is, given $(g, g^a, g^b) \in \mathbb{G}^3$, where $a, b \xleftarrow{\$} \mathbb{Z}_p^*$, to find a triple (C, C^a, C^{ab}) such that $C \neq 1_{\mathbb{G}}$.*

A potentially easier problem considered in [33] only requires to output (C, C^{ab}) on input of the same values. The latter problem was proved generically hard in prime order groups [33]. In bilinear groups, any algorithm solving either of these two problems would make it easy to recognize g^{abc} on input of (g, g^a, g^b, g^c), which is a problem suggested for the first time in [8, Section 8].

2.2 Model and Security Notions

Group encryption schemes involve a sender, a verifier, a group manager (GM) that manages the group of receivers and an opening authority (OA) that is able to uncover the identity of ciphertext receivers. A group encryption system is formally specified by the description of a relation $\mathcal{R}$ as well as a collection $\mathsf{GE} = \big(\mathsf{SETUP}, \mathsf{JOIN}, \langle \mathcal{G}_r, \mathcal{R}, \mathsf{sample}_{\mathcal{R}} \rangle, \mathsf{ENC}, \mathsf{DEC}, \mathsf{OPEN}, \langle \mathcal{P}, \mathcal{V} \rangle\big)$ of algorithms or protocols. Among these, SETUP is a set of initialization procedures that all take (explicitly or implicitly) a security parameter λ as input. They can be split into one that generates a set of public parameters param (a common reference string), one for the GM and another one for the OA. We call them $\mathsf{SETUP}_{\mathsf{init}}(\lambda)$, $\mathsf{SETUP}_{\mathsf{GM}}(\mathsf{param})$ and $\mathsf{SETUP}_{\mathsf{OA}}(\mathsf{param})$, respectively. The latter two procedures are used to produce key pairs $(\mathsf{pk}_{\mathsf{GM}}, \mathsf{sk}_{\mathsf{GM}})$, $(\mathsf{pk}_{\mathsf{OA}}, \mathsf{sk}_{\mathsf{OA}})$ for the GM and the OA. In the following, param is incorporated in the inputs of all algorithms although we sometimes omit to explicitly write it.

$\mathsf{JOIN} = (\mathsf{J}_{\mathsf{user}}, \mathsf{J}_{\mathsf{GM}})$ is an interactive protocol between the GM and the prospective user. As in [30], we will restrict this protocol to have minimal interaction and consist of only two messages: the first one is the user's public key pk sent by $\mathsf{J}_{\mathsf{user}}$ to J_{GM} and the latter's response is a certificate $\mathsf{cert}_{\mathsf{pk}}$ for pk that makes the user's group membership effective. We do not require the user to prove knowledge of his private key sk or anything else about it. In our construction, valid keys will be publicly recognizable and users do not need to prove their validity. After the execution of JOIN, the GM stores the public key pk and its certificate $\mathsf{cert}_{\mathsf{pk}}$ in a public directory $\mathsf{database}$.

Algorithm sample allows sampling pairs $(x, w) \in \mathcal{R}$ (made of a public value x and a witness w) using keys $(\mathsf{pk}_{\mathcal{R}}, \mathsf{sk}_{\mathcal{R}})$ produced by $\mathcal{G}_r$. Depending on the relation, $\mathsf{sk}_{\mathcal{R}}$ may be the empty string (as will be the case in our scheme). The testing procedure $\mathcal{R}(x, w)$ returns 1 whenever $(x, w) \in \mathcal{R}$. To encrypt a witness w such that $(x, w) \in \mathcal{R}$ for some public x, the sender fetches the pair $(\mathsf{pk}, \mathsf{cert}_{\mathsf{pk}})$ from $\mathsf{database}$ and runs the randomized encryption algorithm. The latter takes as input w, a label L, the receiver's pair $(\mathsf{pk}, \mathsf{cert}_{\mathsf{pk}})$ as well as public keys $\mathsf{pk}_{\mathsf{GM}}$ and $\mathsf{pk}_{\mathsf{OA}}$. Its output is a ciphertext $\psi \leftarrow \mathsf{ENC}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}_{\mathsf{pk}}, w, L)$. On input of the same elements, the certificate $\mathsf{cert}_{\mathsf{pk}}$, the ciphertext ψ and the random coins $coins_\psi$ that were used to produce it, the non-interactive algorithm $\mathcal{P}$ generates a proof π_ψ that there exists a certified receiver whose public key was registered in $\mathsf{database}$ and that is able to decrypt ψ and obtain a witness w such that $(x, w) \in \mathcal{R}$. The verification algorithm $\mathcal{V}$ takes as input ψ, $\mathsf{pk}_{\mathsf{GM}}$, $\mathsf{pk}_{\mathsf{OA}}$, π_ψ and the description of $\mathcal{R}$ and outputs 0 or 1. Given ψ, L and the receiver's private key sk, the output of DEC is either a witness w such that $(x, w) \in \mathcal{R}$ or a rejection symbol $\perp$. Finally, OPEN takes as input a ciphertext/label pair (ψ, L) and the OA's secret key $\mathsf{sk}_{\mathsf{OA}}$ and returns a receiver's public key pk.

The security model considers four properties termed correctness, message security, anonymity and soundness. In the definitions hereafter, we sometimes use the notation $\langle \mathsf{output}_A | \mathsf{output}_B \rangle \leftarrow \langle A(\mathsf{input}_A), B(\mathsf{input}_B) \rangle(\mathsf{common\text{-}input})$ to denote the execution of a protocol between A and B obtaining their own outputs from their respective inputs.

Correctness. The correctness property requires that the following experiment returns 1 with overwhelming probability.

```
Experiment $\mathbf{Expt}^{\mathrm{correctness}}(\lambda)$
    $\mathsf{param} \leftarrow \mathsf{SETUP}_{\mathsf{init}}(\lambda)$; $(\mathsf{pk}_{\mathcal{R}}, \mathsf{sk}_{\mathcal{R}}) \leftarrow \mathcal{G}_r(\lambda)$; $(x, w) \leftarrow \mathsf{sample}_{\mathcal{R}}(\mathsf{pk}_{\mathcal{R}}, \mathsf{sk}_{\mathcal{R}})$;
    $(\mathsf{pk}_{\mathsf{GM}}, \mathsf{sk}_{\mathsf{GM}}) \leftarrow \mathsf{SETUP}_{\mathsf{GM}}(\mathsf{param})$; $(\mathsf{pk}_{\mathsf{OA}}, \mathsf{sk}_{\mathsf{OA}}) \leftarrow \mathsf{SETUP}_{\mathsf{OA}}(\mathsf{param})$;
    $\langle \mathsf{pk}, \mathsf{sk}, \mathsf{cert}_{\mathsf{pk}} | \mathsf{pk}, \mathsf{cert}_{\mathsf{pk}} \rangle \leftarrow \langle \mathsf{J}_{\mathsf{user}}, \mathsf{J}_{\mathsf{GM}}(\mathsf{sk}_{\mathsf{GM}}) \rangle(\mathsf{pk}_{\mathsf{GM}})$;
    $\psi \leftarrow \mathsf{ENC}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}_{\mathsf{pk}}, w, L)$;
    $\pi_\psi \leftarrow \mathcal{P}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}, w, L, \psi, coins_\psi)$;
    If $\big((w \neq \mathsf{DEC}(\mathsf{sk}, \psi, L)) \vee (\mathsf{pk} \neq \mathsf{OPEN}(\mathsf{sk}_{\mathsf{OA}}, \psi, L))$
        $\vee(\mathcal{V}(\psi, L, \pi_\psi, \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}) = 0)\big)$ return 0 else return 1;
```

Message Security. The message secrecy property is defined by an experiment where the adversary has access to oracles that may be stateful (and maintain a state across queries) or stateless:

- $\mathsf{DEC}(\mathsf{sk})$: is a stateless oracle for the user decryption function DEC. When this oracle is restricted not to decrypt a ciphertext-label pair (ψ, L), we denote it by $\mathsf{DEC}^{\neg\langle\psi,L\rangle}$.
- $\mathsf{CH}^b_{\mathsf{ror}}(\lambda, \mathsf{pk}, w, L)$: is a real-or-random challenge oracle that is only queried once. It returns $(\psi, coins_\psi)$ such that $\psi \leftarrow \mathsf{ENC}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}_{\mathsf{pk}}, w, L)$ if $b = 1$ whereas, if $b = 0$, $\psi \leftarrow \mathsf{ENC}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}_{\mathsf{pk}}, w', L)$ encrypts a random plaintext uniformly chosen in the space of plaintexts of length $O(\lambda)$. In either case, $coins_\psi$ are the random coins used to generate ψ.
- $\mathsf{PROVE}^b_{\mathcal{P},\mathcal{P}'}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}_{\mathsf{pk}}, \mathsf{pk}_{\mathcal{R}}, x, w, \psi, L, coins_\psi)$: is a stateful oracle that the adversary can query on multiple occasions. If $b = 1$, it runs the real prover $\mathcal{P}$ on the inputs to produce an actual proof π_ψ. If $b = 0$, the oracle runs a simulator $\mathcal{P}'$ that uses the same inputs as $\mathcal{P}$ except witness $w, coins_\psi$ and generates a simulated proof.

These oracles are used in an experiment where the adversary controls the GM, the OA and all members but the honest receiver. The adversary $\mathcal{A}$ is the dishonest GM that certifies the honest receiver in an execution of JOIN. She has oracle access to the decryption function DEC of that receiver. At the challenge phase, she probes the challenge oracle for a label and a pair $(x, w) \in \mathcal{R}$ of her choice. After the challenge phase, she can also invoke the PROVE oracle on multiple occasions and eventually aims to guess the bit b chosen by the challenger.

As pointed out in [29], designing an efficient simulator $\mathcal{P}'$ (for executing $\mathsf{PROVE}^b_{\mathcal{P},\mathcal{P}'}(.)$ when $b = 0$) is part of the security proof and might require a simulated common reference string.

Definition 5. *A GE scheme satisfies message security if, for any PPT adversary $\mathcal{A}$, the experiment below returns* 1 *with probability at most* $1/2 + \mathsf{negl}(\lambda)$.

```
Experiment $\mathbf{Expt}_{\mathcal{A}}^{\mathrm{sec}}(\lambda)$
    $\mathsf{param} \leftarrow \mathsf{SETUP_{init}}(\lambda)$; $(\mathsf{aux}, \mathsf{pk_{GM}}, \mathsf{pk_{OA}}) \leftarrow \mathcal{A}(\mathsf{param})$;
    $\langle \mathsf{pk}, \mathsf{sk}, \mathsf{cert_{pk}} | \mathsf{aux} \rangle \leftarrow \langle \mathsf{J_{user}}, \mathcal{A}(\mathsf{aux}) \rangle (\mathsf{pk_{GM}})$;
    $(\mathsf{aux}, x, w, L, \mathsf{pk}_{\mathcal{R}}) \leftarrow \mathcal{A}^{\mathsf{DEC}(\mathsf{sk},.)}(\mathsf{aux})$; If $(x, w) \notin \mathcal{R}$ return $0$;
    $b \xleftarrow{\$} \{0,1\}$; $(\psi, coins_{\psi}) \leftarrow \mathsf{CH}^{b}_{\mathsf{ror}}(\lambda, \mathsf{pk}, w, L)$;
    $b' \leftarrow \mathcal{A}^{\mathsf{PROVE}^{b}_{\mathcal{P},\mathcal{P}'}(\mathsf{pk_{GM}},\mathsf{pk_{OA}},\mathsf{pk},\mathsf{cert_{pk}},\mathsf{pk}_{\mathcal{R}},x,w,\psi,L,coins_{\psi}),\mathsf{DEC}^{\neg\langle\psi,L\rangle}(\mathsf{sk},.)}(\mathsf{aux}, \psi)$;
    If $b = b'$ return $1$ else return $0$;
```

Anonymity. In anonymity attacks, the adversary controls the whole system but the opening authority and performs a kind of chosen-ciphertext attack on the encryption scheme of the OA. She registers two keys $\mathsf{pk}_0, \mathsf{pk}_1$ in $\mathsf{database}$ and, for a pair $(x, w) \in \mathcal{R}$ of her choosing, obtains an encryption of w under pk_b for some $b \in \{0, 1\}$ chosen by the challenger. She is granted access to decryption oracles w.r.t. both keys $\mathsf{pk}_0, \mathsf{pk}_1$. In addition, she may invoke the following oracles:

- $\mathsf{CH}^{b}_{\mathsf{anon}}(\mathsf{pk_{GM}}, \mathsf{pk_{OA}}, \mathsf{pk}_0, \mathsf{pk}_1, w, L)$: is a challenge oracle that is only queried once by the adversary. It returns a pair $(\psi, coins_{\psi})$ consisting of a ciphertext $\psi \leftarrow \mathsf{ENC}(\mathsf{pk_{GM}}, \mathsf{pk_{OA}}, \mathsf{pk}_b, \mathsf{cert}_{\mathsf{pk}_b}, w, L)$ and the coin tosses $coins_{\psi}$ that were used to generate ψ.
- $\mathsf{USER}(\mathsf{pk_{GM}})$: is a stateful oracle simulating two executions of $\mathsf{J_{user}}$ to introduce two honest users in the group. It uses a string keys where the outputs of the two executions are written.
- $\mathsf{OPEN}(\mathsf{sk_{OA}}, .)$: is a stateless oracle that simulates the opening algorithm on behalf of the OA and, on input of a GE ciphertext, returns the receiver's public key.

Definition 6. *A GE scheme satisfies anonymity if, for any PPT adversary $\mathcal{A}$, the experiment below returns 1 with a probability not exceeding $1/2 + \mathsf{negl}(\lambda)$.*

```
Experiment $\mathbf{Expt}_{\mathcal{A}}^{\mathrm{anon}}(\lambda)$
    $\mathsf{param} \leftarrow \mathsf{SETUP_{init}}(\lambda)$; $(\mathsf{pk_{OA}}, \mathsf{sk_{OA}}) \leftarrow \mathsf{SETUP_{OA}}(\mathsf{param})$;
    $(\mathsf{aux}, \mathsf{pk_{GM}}) \leftarrow \mathcal{A}(\mathsf{param}, \mathsf{pk_{OA}})$; $\mathsf{aux} \leftarrow \mathcal{A}^{\mathsf{USER}(\mathsf{pk_{GM}}),\mathsf{OPEN}(\mathsf{sk_{OA}},.)}(\mathsf{aux})$;
    If $\mathsf{keys} \neq (\mathsf{pk}_0, \mathsf{sk}_0, \mathsf{cert}_{\mathsf{pk}_0}, \mathsf{pk}_1, \mathsf{sk}_1, \mathsf{cert}_{\mathsf{pk}_1})(\mathsf{aux})$ return $0$;
    $(\mathsf{aux}, x, w, L, \mathsf{pk}_{\mathcal{R}}) \leftarrow \mathcal{A}^{\mathsf{OPEN}(\mathsf{sk_{OA}},.),\mathsf{DEC}(\mathsf{sk}_0,.),\mathsf{DEC}(\mathsf{sk}_1,.)}(\mathsf{aux})$;
    If $(x, w) \notin \mathcal{R}$ return $0$;
    $b \xleftarrow{\$} \{0,1\}$; $(\psi, coins_{\psi}) \leftarrow \mathsf{CH}^{b}_{\mathsf{anon}}(\mathsf{pk_{GM}}, \mathsf{pk_{OA}}, \mathsf{pk}_0, \mathsf{pk}_1, w, L)$;
    $b' \leftarrow \mathcal{A}^{\mathcal{P}(\mathsf{pk_{GM}},\mathsf{pk_{OA}},\mathsf{pk}_b,\mathsf{cert}_{\mathsf{pk}_b},x,w,\psi,L,coins_{\psi},$
                    $\mathsf{OPEN}^{\neg\langle\psi,L\rangle}(\mathsf{sk_{OA}},.),\mathsf{DEC}^{\neg\langle\psi,L\rangle}(\mathsf{sk}_0,.),\mathsf{DEC}^{\neg\langle\psi,L\rangle}(\mathsf{sk}_1,.))}(\mathsf{aux}, \psi)$;
    If $b = b'$ return $1$ else return $0$;
```

As shown in [29], GE schemes satisfying the above notion necessarily subsume a key-private (a.k.a. receiver anonymous) [3, 28] cryptosystem.

Soundness. In a soundness attack, the adversary creates the group of receivers by interacting with the honest GM. Her goal is to produce a ciphertext ψ and a convincing proof that ψ is valid w.r.t. a relation $\mathcal{R}$ of her choice but either (1) the opening reveals a receiver's public key pk that does not belong to any group member; (2) the output pk of OPEN is not a valid public key (*i.e.*, $\mathsf{pk} \notin \mathcal{PK}$, where $\mathcal{PK}$ is the space of valid public keys); (3) the ciphertext C is not in the space $\mathcal{C}^{x,L,\mathsf{pk}_{\mathcal{R}},\mathsf{pk_{GM}},\mathsf{pk_{OA}},\mathsf{pk}}$ of valid ciphertexts. This notion is formalized by a game where the adversary is given access to a user registration oracle $\mathsf{REG}(\mathsf{sk_{GM}}, .)$ that simulates $\mathsf{J_{GM}}$. This oracle maintains a repository $\mathsf{database}$ where registered public keys and their certificates are stored.

Definition 7. *A GE scheme is sound if, for any PPT adversary $\mathcal{A}$, the experiment below returns* 1 *with negligible probability.*

```
Experiment $\mathbf{Expt}_{\mathcal{A}}^{\mathrm{soundness}}(\lambda)$
    $\mathsf{param} \leftarrow \mathsf{SETUP_{init}}(\lambda)$; $(\mathsf{pk_{OA}}, \mathsf{sk_{OA}}) \leftarrow \mathsf{SETUP_{OA}}(\mathsf{param})$;
    $(\mathsf{pk_{GM}}, \mathsf{sk_{GM}}) \leftarrow \mathsf{SETUP_{GM}}(\mathsf{param})$;
    $(\mathsf{pk}_{\mathcal{R}}, x, \psi, \pi_\psi, L, \mathsf{aux}) \leftarrow \mathcal{A}^{\mathsf{REG}(\mathsf{sk_{GM}},\cdot)}(\mathsf{param}, \mathsf{pk_{GM}}, \mathsf{pk_{OA}}, \mathsf{sk_{OA}})$;
    If $\mathcal{V}(\psi, L, \pi_\psi, \mathsf{pk_{GM}}, \mathsf{pk_{OA}}) = 0$ return 0;
    $\mathsf{pk} \leftarrow \mathsf{OPEN}(\mathsf{sk_{OA}}, \psi, L)$;
    If $\big((\mathsf{pk} \notin \mathsf{database}) \vee (\mathsf{pk} \notin \mathcal{PK}) \vee (\psi \notin \mathcal{C}^{x, L, \mathsf{pk}_{\mathcal{R}}, \mathsf{pk_{GM}}, \mathsf{pk_{OA}}, \mathsf{pk}})\big)$
        then return 1 else return 0;
```

2.3 Groth-Sahai Proof Systems

In the following notations, for equal-dimension vectors $\vec{A}$ and $\vec{B}$ containing group elements, $\vec{A} \odot \vec{B}$ stands for their component-wise product.

When based on the DLIN assumption, the Groth-Sahai (GS) proof systems [27] use a common reference string comprising vectors $\vec{g_1}, \vec{g_2}, \vec{g_3} \in \mathbb{G}^3$, where $\vec{g_1} = (g_1, 1, g)$, $\vec{g_2} = (1, g_2, g)$ for some $g_1, g_2 \in \mathbb{G}$. To commit to $X \in \mathbb{G}$, one sets $\vec{C} = (1, 1, X) \odot \vec{g_1}^{\,r} \odot \vec{g_2}^{\,s} \odot \vec{g_3}^{\,t}$ with $r, s, t \xleftarrow{\$} \mathbb{Z}_p^*$. When the proof system is prepared to give perfectly sound proofs, $\vec{g_3}$ is set as $\vec{g_3} = \vec{g_1}^{\,\xi_1} \odot \vec{g_2}^{\,\xi_2}$ with $\xi_1, \xi_2 \xleftarrow{\$} \mathbb{Z}_p^*$. Commitments $\vec{C} = (g_1^{r+\xi_1 t}, g_2^{s+\xi_2 t}, X \cdot g^{r+s+t(\xi_1+\xi_2)})$ are then Boneh-Boyen-Shacham (BBS) ciphertexts that can be decrypted using $\alpha_1 = \log_g(g_1)$, $\alpha_2 = \log_g(g_2)$. In the witness indistinguishability (WI) setting, vectors $\vec{g_1}, \vec{g_2}, \vec{g_3}$ are linearly independent and $\vec{C}$ is a perfectly hiding commitment. Under the DLIN assumption, the two kinds of CRS are indistinguishable.

To commit to an exponent $x \in \mathbb{Z}_p$, one computes $\vec{C} = \vec{\varphi}^{\,x} \odot \vec{g_1}^{\,r} \odot \vec{g_2}^{\,s}$, with $r, s \xleftarrow{\$} \mathbb{Z}_p^*$, using a CRS comprising vectors $\vec{\varphi}, \vec{g_1}, \vec{g_2}$. In the soundness setting $\vec{\varphi}, \vec{g_1}, \vec{g_2}$ are linearly independent vectors (typically $\vec{\varphi} = \vec{g_3} \odot (1, 1, g)$ where $\vec{g_3} = \vec{g_1}^{\,\xi_1} \odot \vec{g_2}^{\,\xi_2}$) whereas, in the WI setting, choosing $\vec{\varphi} = \vec{g_1}^{\,\xi_1} \odot \vec{g_2}^{\,\xi_2}$ gives a perfectly hiding commitment since $\vec{C}$ is always a BBS encryption of $1_{\mathbb{G}}$.

To prove that committed variables satisfy a set of relations, the GS techniques replace variables by the corresponding commitments in each relation. The whole proof consists of one commitment per variable and one proof element (made of a constant number of group elements) per relation.

Such proofs are available for pairing-product relations, which are of the type

$$\prod_{i=1}^{n} e(\mathcal{A}_i, \mathcal{X}_i) \cdot \prod_{i=1}^{n} \cdot \prod_{j=1}^{n} e(\mathcal{X}_i, \mathcal{X}_j)^{a_{ij}} = t_T,$$

for variables $\mathcal{X}_1, \ldots, \mathcal{X}_n \in \mathbb{G}$ and constants $t_T \in \mathbb{G}_T$, $\mathcal{A}_1, \ldots, \mathcal{A}_n \in \mathbb{G}$, $a_{ij} \in \mathbb{G}$, for $i, j \in \{1, \ldots, n\}$. Efficient proofs also exist for multi-exponentiation equations

$$\prod_{i=1}^{m} \mathcal{A}_i^{y_i} \cdot \prod_{j=1}^{n} \mathcal{X}_j^{b_j} \cdot \prod_{i=1}^{m} \cdot \prod_{j=1}^{n} \mathcal{X}_j^{y_i \gamma_{ij}} = T,$$

for variables $\mathcal{X}_1, \ldots, \mathcal{X}_n \in \mathbb{G}$, $y_1, \ldots, y_m \in \mathbb{Z}_p$ and constants $T, \mathcal{A}_1, \ldots, \mathcal{A}_m \in \mathbb{G}$, $b_1, \ldots, b_n \in \mathbb{Z}_p$ and $\gamma_{ij} \in \mathbb{G}$, for $i \in \{1, \ldots, m\}, j \in \{1, \ldots, n\}$.

Multi-exponentiation equations admit zero-knowledge proofs at no additional cost. On a simulated CRS (prepared for the WI setting), a trapdoor makes it is possible to simulate proofs without knowing witnesses and simulated proofs are perfectly indistinguishable from real proofs. As for pairing-product equations, zero-knowledge proofs are often possible but usually come at some expense. In the paper, we only resort to such NIZK simulators in one occasion.

In both cases, proofs for quadratic equations cost 9 group elements. Linear pairing-product equations (when $a_{ij} = 0$ for all i, j) take 3 group elements each. Linear multi-exponentiation equations of the type $\prod_{j=1}^{n} \mathcal{X}_j^{b_j} = T$ (resp. $\prod_{i=1}^{m} \mathcal{A}_i^{y_i} = T$) demand 3 (resp. 2) group elements.

3 Building Blocks

Our certification scheme uses a trapdoor commitment to group elements as an important ingredient to dispense with proofs of knowledge of users' private keys.

3.1 A Trapdoor Commitment to Group Elements

We need a trapdoor commitment scheme that allows committing to elements of a group $\mathbb{G}$ where bilinear map arguments are taken. Commitments will have to be themselves elements of $\mathbb{G}$, which prevents us from using Groth's scheme [25] where commitments lie in the range $\mathbb{G}_T$ of the pairing.

Such commitments can be obtained using the perfectly hiding Groth-Sahai commitment based on the linear assumption recalled in section 2.3. This commitment uses a common reference string describing a prime order group $\mathbb{G}$ and a generator $f \in \mathbb{G}$. The commitment key consists of vectors $(\vec{f_1}, \vec{f_2}, \vec{f_3})$ chosen as $\vec{f_1} = (f_1, 1, f)$, $\vec{f_2} = (1, f_2, f)$ and $\vec{f_3} = \vec{f_1}^{\xi_1} \odot \vec{f_2}^{\xi_2} \odot (1, 1, f)^{\xi_3}$, with $f_1, f_2 \stackrel{\$}{\leftarrow} \mathbb{G}$, $\xi_1, \xi_2, \xi_3 \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$. To commit to a group element $X \in \mathbb{G}$, the sender picks $\phi_1, \phi_2, \phi_3 \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$ and sets $\vec{C}_X = (1, 1, X) \odot \vec{f_1}^{\phi_1} \odot \vec{f_2}^{\phi_2} \odot \vec{f_3}^{\phi_3}$, which, if $\vec{f_3}$ is parsed as $(f_{3,1}, f_{3,2}, f_{3,3})$, can be written $\vec{C}_X = (f_1^{\phi_1} \cdot f_{3,1}^{\phi_3}, f_2^{\phi_2} \cdot f_{3,2}^{\phi_3}, X \cdot f^{\phi_1+\phi_2} \cdot f_{3,3}^{\phi_3})$. Due to the use of GS proofs, commitment openings need to only consist of group elements (and no scalar). To open $\vec{C}_X = (C_1, C_2, C_3)$, the sender reveals $(D_1, D_2, D_3) = (f^{\phi_1}, f^{\phi_2}, f^{\phi_3})$ and X. The receiver is convinced that the committed value was X by checking that

$$\begin{cases} e(C_1, f) = e(f_1, D_1) \cdot e(f_{3,1}, D_3) \\ e(C_2, f) = e(f_2, D_2) \cdot e(f_{3,2}, D_3) \\ e(C_3, f) = e(X \cdot D_1 \cdot D_2, f) \cdot e(f_{3,3}, D_3). \end{cases}$$

If a cheating sender can come up with distinct openings of $\vec{C}_X$, we can easily solve a $S2P$ instance $(g_1, g_2, g_{1,c}, g_{2,d})$. Namely, the commitment key is set as $(f_1, f_2, f_{3,1}, f_{3,2}) = (g_1, g_2, g_{1,c}, g_{2,d})$ and $f, f_{3,3}$ are chosen at random. When the adversary outputs $(X, (D_1, D_2, D_3))$ and $(X', (D_1', D_2', D_3'))$, we must simultaneously have $e(f_1, D_1/D_1') = e(f_{3,1}, D_3'/D_3)$, $e(f_2, D_2/D_2') = e(f_{3,2}, D_3'/D_3)$ and $e((XD_1D_2)/(X'D_1'D_2'), f) = e(f_{3,3}, D_3'/D_3)$. Hence, a solution to the S2P instance is obtained by setting $(u, v, w) = (D_1/D_1', D_2/D_2', D_3'/D_3)$, which is a non-trivial triple as long as $X' \neq X$.

We also observe that, using the trapdoor (ξ_1, ξ_2, ξ_3), the receiver can equivocate commitments. Given a commitment $\vec{C}_X$ and its opening $(X, (D_1, D_2, D_3))$, one can trapdoor open $\vec{C}_X$ to any other $X' \in \mathbb{G}$ (and without knowing $\log_g(X')$) by computing

$$D_1' = D_1 \cdot (X'/X)^{\xi_1/\xi_3}, \qquad D_2' = D_2 \cdot (X'/X)^{\xi_2/\xi_3}, \qquad D_3' = (X/X')^{1/\xi_3} \cdot D_3.$$

3.2 A Public Key Certification Scheme

We use a primitive that we call *non-interactive certification scheme*, which can be viewed as a signature scheme that only allows signing public keys from a specific public key space $\mathcal{PK}$. These keys should be signed while retaining algebraic properties that make it possible to prove knowledge of a public key and its corresponding certificate in an efficient way. In particular, signing hashed public keys is proscribed. In the interactive setting, several papers (e.g., [5, 24]) describe efficient interactive protocols where a public key is jointly generated by a user and a certification authority in such a way that the user eventually obtains a certified public key and no one else learns the underlying private key. In this paper, we aim at minimizing the amount of interaction and let users generate their public key entirely on their own before requesting their certification. Ideally, we would like to be able to sign public keys without even requiring users to prove knowledge of their private key and, in particular, without having to first rewind a proof of knowledge so as to extract the user's private key in the security proof.

A certification scheme consists of algorithms $(\mathsf{Setup}, \mathsf{Certify}, \mathsf{CertVerify})$. The first one is run by a certification authority (CA) that, on input of global parameters cp, generates a key pair $(SK, PK) \leftarrow \mathsf{Setup}(\mathsf{cp})$. On input of cp, SK and a user's public key pk, $\mathsf{Certify}$ generates a certificate $\mathsf{cert}_{\mathsf{pk}}$. The procedure Verify takes as input cp, PK, pk and $\mathsf{cert}_{\mathsf{pk}}$ and outputs either 0 or 1.

Correctness mandates that $\mathsf{CertVerify}(\mathsf{cp}, PK, \mathsf{pk}, \mathsf{cert}_{\mathsf{pk}}) = 1$ when $\mathsf{cert}_{\mathsf{pk}} \leftarrow \mathsf{Certify}(\mathsf{cp}, SK, \mathsf{pk})$. The (strong) unforgeability [1] requirement is the same as in signature schemes. The adversary is supplied with a CA's public key PK and access to a certification oracle $\mathsf{Certify}(SK, .)$ that can be queried for arbitrary public keys $\mathsf{pk} \in \mathcal{PK}$. Her goal is to produce a new pair $(\mathsf{pk}^*, \mathsf{cert}^*_{\mathsf{pk}^*})$ (*i.e.*, if pk^* was queried to $\mathsf{Certify}(SK, .)$, the output must have been different from $\mathsf{cert}^*_{\mathsf{pk}^*}$).

In the description, we assume common public parameters cp consisting of of bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$, for a security parameter λ, and a generator $g \xleftarrow{\$} \mathbb{G}$. We also assume that certified public keys always consist of a fixed number n of group elements (*i.e.*, $\mathcal{PK} = \mathbb{G}^n$).

Intuition. The scheme borrows from the Boyen-Waters group signature [10] in the use of the Hidden Strong Diffie-Hellman assumption. A simplified version of this scheme involves a CA that holds a public key $PK = (\Omega = g^\omega, A = (g,g)^\alpha, u, u_0, u_1 = g^{\beta_1}, \dots, u_n = g^{\beta_n})$, for private elements $SK = (\omega, \alpha, \beta_1, \dots, \beta_n)$, where n denotes the number of groups elements that certified public keys consist of. To certify a public key $\mathsf{pk} = (X_1 = g^{x_1}, \dots, X_n = g^{x_n})$, the CA chooses an exponent $c_{\mathsf{ID}} \xleftarrow{\$} \mathbb{Z}_p^*$ and computes $S_1 = (g^\alpha)^{1/(\omega + c_{\mathsf{ID}})}$, $S_2 = g^{c_{\mathsf{ID}}}$, $S_3 = u^{c_{\mathsf{ID}}}$, $S_4 = (u_0 \cdot \prod_{i=1}^n X_i^{\beta_i})^{c_{\mathsf{ID}}}$ and $S_5 = (S_{5,1}, \dots, S_{5,n}) = (X_1^{c_{\mathsf{ID}}}, \dots, X_n^{c_{\mathsf{ID}}})$. Verification then checks whether $e(S_1, \Omega \cdot S_2) = A$ and $e(S_2, u) = e(g, S_3)$ as in [10]. It must also be checked that $e(S_4, g) = e(u_0, S_2) \cdot \prod_{i=1}^n e(u_i, S_{5,i})$ and $e(S_{5,i}, g) = e(X_i, S_2)$ for $i = 1, \dots, n$.

The security of this simplified scheme can only be proven if, when answering certification queries, the simulator can control the private keys $(x_1, \dots, x_n)$ and force them to be random values of its choice. To allow the simulator to sign arbitrary public keys without knowing the private keys, we modify the scheme so that the CA rather signs commitments (calculated as in the trapdoor commitment of section 3.1) to public key elements $X_1, \dots, X_n$. In the security proof, the simulator first generates a signature on n fake commitments $\vec{C}_i = (C_{i,1}, C_{i,2}, C_{i,3})$ that are all generated in such a way that it knows $\log_g(C_{i,j})$ for $i = 1, \dots, n$ and $j = 1, 2, 3$. Using the trapdoor of the commitment scheme, it can then open $\vec{C}_i$ to any arbitrary $X_i \in \mathbb{G}$ without knowing $\log_g(X_i)$.

This use of the trapdoor commitment is reminiscent of a technique (notably used in [18]) to construct signature schemes in the standard model using chameleon hash functions [32]: the simulator first signs messages of its choice using a basic signature scheme and then “equivocates” the chameleon hashes to make them correspond to adversarially-chosen messages.

Setup(cp)**:** given common public parameters $\mathsf{cp} = \{g, \mathbb{G}, \mathbb{G}_T\}$, select $u, u_0 \xleftarrow{\$} \mathbb{G}$, $\alpha, \omega \xleftarrow{\$} \mathbb{Z}_p^*$ and set $A = e(g,g)^\alpha$, $\Omega = g^\omega$. Pick $\beta_{i,1}, \beta_{i,2}, \beta_{i,3} \xleftarrow{\$} \mathbb{Z}_p^*$ and set $\overline{u}_i = (u_{i,1}, u_{i,2}, u_{i,3}) = (g^{\beta_{i,1}}, g^{\beta_{i,2}}, g^{\beta_{i,3}})$ for $i = 1, \ldots, n$. Choose $f, f_1, f_2, f_{3,1}, f_{3,2}, f_{3,3} \xleftarrow{\$} \mathbb{G}$ that define a commitment key consisting of vectors $\vec{f_1} = (f_1, 1, f)$, $\vec{f_2} = (1, f_2, f)$ and $\vec{f_3} = (f_{3,1}, f_{3,2}, f_{3,3})$. Define the private key to be $SK = \big(\alpha, \omega, \{\overline{\beta}_i = (\beta_{i,1}, \beta_{i,2}, \beta_{i,3})\}_{i=1,\ldots,n}\big)$ and the public key as

$$PK = \Big(\mathbf{f} = (\vec{f_1}, \vec{f_2}, \vec{f_3}),\ A = e(g,g)^\alpha,\ \Omega = g^\omega,\ u,\ u_0,\ \{\overline{u}_i\}_{i=1,\ldots,n}\Big).$$

Certify$(\mathsf{cp}, SK, \mathsf{pk})$**:** parse SK as $\big(\alpha, \omega, \{\overline{\beta}_i\}_{i=1,\ldots,n}\big)$, pk as $(X_1, \ldots, X_n)$ and do the following.

1. For each $i \in \{1, \ldots, n\}$, pick $\phi_{i,1}, \phi_{i,2}, \phi_{i,3} \xleftarrow{\$} \mathbb{Z}_p^*$ and compute a commitment

$$C_i = (C_{i,1}, C_{i,2}, C_{i,3}) = (f_1^{\phi_{i,1}} \cdot f_{3,1}^{\phi_{i,3}},\ f_2^{\phi_{i,2}} \cdot f_{3,2}^{\phi_{i,3}},\ X_i \cdot f^{\phi_{i,1}+\phi_{i,2}} \cdot f_{3,3}^{\phi_{i,3}})$$

and the matching de-commitment $(D_{i,1}, D_{i,2}, D_{i,3}) = (f^{\phi_{i,1}}, f^{\phi_{i,2}}, f^{\phi_{i,3}})$.

2. Choose $c_{\mathsf{ID}} \xleftarrow{\$} \mathbb{Z}_p^*$ and compute $S_1 = (g^\alpha)^{1/(\omega + c_{\mathsf{ID}})}$, $S_2 = g^{c_{\mathsf{ID}}}$, $S_3 = u^{c_{\mathsf{ID}}}$ as well as

$$S_4 = \Big(u_0 \cdot \prod_{i=1}^{n} (C_{i,1}^{\beta_{i,1}} \cdot C_{i,2}^{\beta_{i,2}} \cdot C_{i,3}^{\beta_{i,3}})\Big)^{c_{\mathsf{ID}}}$$
$$S_5 = \{(S_{5,i,1}, S_{5,i,2}, S_{5,i,3})\}_{i=1,\ldots,n} = \{(C_{i,1}^{c_{\mathsf{ID}}}, C_{i,2}^{c_{\mathsf{ID}}}, C_{i,3}^{c_{\mathsf{ID}}})\}_{i=1,\ldots,n}$$

Return $\mathsf{cert}_{\mathsf{pk}} = \Big(\{(C_{i,1}, C_{i,2}, C_{i,3}), (D_{i,1}, D_{i,2}, D_{i,3})\}_{i=1,\ldots,n}, S_1, S_2, S_3, S_4, S_5\Big)$.

CertVerify$(\mathsf{cp}, PK, \mathsf{pk}, \mathsf{cert}_{\mathsf{pk}})$**:** parse pk as $(X_1, \ldots, X_n)$ and $\mathsf{cert}_{\mathsf{pk}}$ as above. Return 1 if, for indices $i = 1, \ldots, n$, it holds that $X_i \in \mathbb{G}$ and

$$e(C_{i,1}, f) = e(f_1, D_{i,1}) \cdot e(f_{3,1}, D_{i,3}) \tag{1}$$
$$e(C_{i,2}, f) = e(f_2, D_{i,2}) \cdot e(f_{3,2}, D_{i,3}) \tag{2}$$
$$e(C_{i,3}, f) = e(X_i \cdot D_{i,1} \cdot D_{i,2}, f) \cdot e(f_{3,3}, D_{i,3}), \tag{3}$$

and if the following checks are also satisfied. Otherwise, return 0.

$$e(S_1, \Omega \cdot S_2) = A \tag{4}$$
$$e(S_2, u) = e(g, S_3) \tag{5}$$
$$e(S_4, g) = e(u_0, S_2) \cdot \prod_{i=1}^{n} \big(e(u_{i,1}, S_{5,i,1}) \cdot e(u_{i,2}, S_{5,i,2}) \cdot e(u_{i,3}, S_{5,i,3})\big), \tag{6}$$
$$e(S_{5,i,j}, g) = e(C_{i,j}, S_2) \qquad \text{for } i = 1, \ldots, n,\ j = 1,2,3 \tag{7}$$

A certificate comprises $9n+4$ group elements. It would be interesting to avoid this linear dependency on n without destroying the algebraic properties that render the scheme compatible with the Groth-Sahai techniques.

Regarding the security of the scheme, the following theorem is proved in appendix A.

Theorem 1. *The scheme is a secure non-interactive certification system if the HSDH, FlexDH and S2P problems are all hard in $\mathbb{G}$.*

We believe that the above certification scheme is of interest in its own right. For instance, it can be used to construct non-frameable group signatures that are secure in the concurrent join model of [30] without resorting to random oracles. To the best of our knowledge, the Kiayias-Yung construction [30] has remained the only scalable group signature where joining supports concurrency at both ends while requiring the smallest amount of interaction. In the standard model, our certification scheme thus appears to provide the first[4] way to achieve the same result. In this case, we have $n = 1$ (since prospective group members only need to certify one group element if non-frameability is ensured by signing messages using Boneh-Boyen signatures [6] in the same way as in Groth's group signature [24]) so that membership certificates comprise 13 group elements and their shape is fully compatible with GS proofs.

3.3 Public Key Encryption Schemes Based on the Linear Problem

We need cryptosystems based on the DLIN assumption. The first one is Shacham's variant [37] of Cramer-Shoup [17] and, since it is key-private [3], we use it to encrypt witnesses. We also use Kiltz's tag-based encryption (TBE) scheme [31], where the validity of ciphertexts is publicly verifiable, to encrypt receivers' public keys under the public key of the opening authority.

Shacham's Linear Cramer-Shoup. If we assume public generators g_1, g_2, g that are parts of public parameters, each receiver's public key is made of $n = 6$ group elements

$$\begin{array}{lll} X_1 = g_1^{x_1} g^x & X_3 = g_1^{x_3} g^y & X_5 = g_1^{x_5} g^z \\ X_2 = g_2^{x_2} g^x & X_4 = g_2^{x_4} g^y & X_6 = g_2^{x_6} g^z. \end{array}$$

To encrypt a plaintext $m \in \mathbb{G}$ under the label L, the sender picks $r, s \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$ and computes

$$\psi_{\mathsf{CS}} = \big(U_1, U_2, U_3, U_4, U_5\big) = \Big(g_1^r,\ g_2^s,\ g^{r+s},\ m \cdot X_5^r X_6^s,\ (X_1 X_3^\alpha)^r \cdot (X_2 X_4^\alpha)^s\Big),$$

where $\alpha = H(U_1, U_2, U_3, U_4, L) \in \mathbb{Z}_p^*$ is a collision-resistant hash[5]. Given (ψ_{CS}, L), the receiver computes α. He returns $\perp$ if $U_5 \neq U_1^{x_1+\alpha x_3} U_2^{x_2+\alpha x_4} U_3^{x+\alpha y}$ and $m = U_4/(U_1^{x_5} U_2^{x_6} U_3^z)$ otherwise.

Kiltz's Tag-Based Encryption Scheme. In [31], Kiltz described a TBE scheme based on the same assumption. The public key is $(Y_1, Y_2, Y_3, Y_4) = (g^{y_1}, g^{y_2}, g^{y_3}, g^{y_4})$ if $g \in \mathbb{G}$ is part of public parameters. To encrypt $m \in \mathbb{G}$ under a tag $t \in \mathbb{Z}_p^*$, the sender picks $w_1, w_2 \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$ and computes

$$\psi_{\mathsf{K}} = (V_1, V_2, V_3, V_4, V_5) = \Big(Y_1^{w_1},\ Y_2^{w_2},\ (g^t Y_3)^{w_1},\ (g^t Y_4)^{w_2},\ m \cdot g^{w_1+w_2}\Big)$$

[4] Non-frameable group signatures described in [19, 9] achieve concurrent security by having the prospective user generate an extractable commitment to some secret exponent (which the simulator can extract without rewinding using the trapdoor of the commitment) and prove that the committed value is the discrete log. of a public value. In the standard model, this technique requires interaction and the proof should be simulatable in zero-knowledge when proving security against framing attacks. Another technique [21] requires users to prove knowledge of their secret exponent using Groth-Sahai non-interactive proofs. It is nevertheless space-demanding as each bit of committed exponent requires its own extractable GS commitment.

[5] The proof of CCA2-security [17, 37] only requires a universal one-way hash function (UOWHF) [34] but collision-resistance is required if the scheme has to support labels.

To decrypt such a ciphertext ψ_{K}, the receiver checks that $V_3 = V_1^{(t+y_3)/y_1}$, $V_4 = V_2^{(t+y_4)/y_2}$. If so, it outputs the plaintext $m = V_5/(V_1^{1/y_1} V_2^{1/y_2})$. Unlike ψ_{CS} in the linear Cramer-Shoup system, the well-formedness of ψ_{K} is publicly verifiable in bilinear groups. The Canetti-Halevi-Katz [15] paradigm turns this scheme into a full-fledged CCA2 scheme by deriving the tag t from the verification key VK of a one-time signature, the private key SK of which is used to sign $(V_1, V_2, V_3, V_4, V_5)$.

4 A GE Scheme with Non-Interactive Proofs

We build a non-interactive group encryption scheme for the Diffie-Hellman relation $\mathcal{R} = \{(X, Y), W\}$ where $e(g, W) = e(X, Y)$, for which the keys are $\mathsf{pk}_{\mathcal{R}} = \{\mathbb{G}, \mathbb{G}_T, g\}$ and $\mathsf{sk}_{\mathcal{R}} = \varepsilon$.

The construction slightly departs from the modular design of [29] in that commitments to the receiver's public key and certificate are part of the proof (instead of the ciphertext), which simplifies the proof of message-security. The security of the scheme eventually relies on the HSDH, FlexDH and DLIN assumptions. All security proofs are available in appendix B.

$\mathsf{SETUP}_{\mathsf{init}}(\lambda)$: choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of order $p > 2^\lambda$, $g \xleftarrow{\$} \mathbb{G}$ and $g_1 = g^{\alpha_1}$, $g_2 = g^{\alpha_2}$ with $\alpha_1, \alpha_2 \xleftarrow{\$} \mathbb{Z}_p^*$. Define $\vec{g_1} = (g_1, 1, g)$, $\vec{g_2} = (1, g_2, g)$ and $\vec{g_3} = \vec{g_1}^{\xi_1} \odot \vec{g_2}^{\xi_2}$ with $\xi_1, \xi_2 \xleftarrow{\$} \mathbb{Z}_p^*$, which form a CRS $\mathbf{g} = (\vec{g_1}, \vec{g_2}, \vec{g_3})$ for the perfect soundness setting. Select a strongly unforgeable (as defined in [1]) one time signature scheme $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ and a random member $H : \{0,1\}^* \to \mathbb{Z}_p$ of a collision-resistant hash family. Public parameters consists of $\mathsf{param} = \{\lambda, \mathbb{G}, \mathbb{G}_T, g, \mathbf{g}, \Sigma, H\}$.

$\mathsf{SETUP}_{\mathsf{GM}}(\mathsf{param})$: runs the setup algorithm of the certification scheme described in section 3.2 with $n = 6$. The obtained public key consists of $\mathsf{pk}_{\mathsf{GM}} = \Big(\mathbf{f},\ A = e(g, g)^\alpha,\ \Omega = g^\omega,\ u,\ u_0,\ \{\overline{u}_i\}_{i=1,\ldots,6}\Big)$ and the matching private key is $\mathsf{sk}_{\mathsf{GM}} = \big(\alpha, \omega, \{\overline{\beta}_i = (\beta_{i,1}, \beta_{i,2}, \beta_{i,3})\}_{i=1,\ldots,6}\big)$.

$\mathsf{SETUP}_{\mathsf{OA}}(\mathsf{param})$: generates $\mathsf{pk}_{\mathsf{OA}} = (Y_1, Y_2, Y_3, Y_4) = (g^{y_1}, g^{y_2}, g^{y_3}, g^{y_4})$, as a public key for Kiltz's tag-based encryption scheme [31], and the corresponding private key as $\mathsf{sk}_{\mathsf{OA}} = (y_1, y_2, y_3, y_4)$.

JOIN: the user sends a linear Cramer-Shoup public key $\mathsf{pk} = (X_1, \ldots, X_6) \in \mathbb{G}^6$ to the GM and obtains a certificate

$$\mathsf{cert}_{\mathsf{pk}} = \big(\{(C_{i,1}, C_{i,2}, C_{i,3}), (D_{i,1}, D_{i,2}, D_{i,3})\}_{i=1,\ldots,6}, S_1, S_2, S_3, S_4, S_5\big).$$

$\mathsf{ENC}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}_{\mathsf{pk}}, W, L)$: to encrypt $W \in \mathbb{G}$ such that $((X, Y), W) \in \mathcal{R}$ (for public elements $X, Y \in \mathbb{G}$), parse $\mathsf{pk}_{\mathsf{GM}}$, $\mathsf{pk}_{\mathsf{OA}}$ and pk as above and do the following.

1. Generate a one-time signature key pair $(\mathsf{SK}, \mathsf{VK}) \leftarrow \mathcal{G}(\lambda)$.
2. Choose $r, s \xleftarrow{\$} \mathbb{Z}_p^*$ and compute a linear CS encryption of W, the result of which is denoted by ψ_{CS}, under the label $L_1 = L||\mathsf{VK}$ as per section 3.3 (and using the collision-resistant hash function specified by param).
3. For $i = 1, \ldots, 6$, choose $w_{i,1}, w_{i,2} \xleftarrow{\$} \mathbb{Z}_p^*$ and encrypt X_i under $\mathsf{pk}_{\mathsf{OA}}$ using Kiltz's TBE with the tag VK as described in section 3.3 . Let ψ_{K_i} be the ciphertexts.
4. Set the GE ciphertext ψ as $\psi = \mathsf{VK}||\psi_{\mathsf{CS}}||\psi_{\mathsf{K}_1}|| \cdots ||\psi_{\mathsf{K}_6}||\sigma$ where σ is a one-time signature obtained as $\sigma = \mathcal{S}(\mathsf{SK}, (\psi_{\mathsf{CS}}||\psi_{\mathsf{K}_1}|| \cdots ||\psi_{\mathsf{K}_6}||L))$.

Return (ψ, L) and $coins_\psi$ consist of $\{(w_{i,1}, w_{i,2})\}_{i=1,\ldots,6}$, (r, s). If the one-time signature of [23] is used, VK and σ take 3 and 2 group elements, respectively, so that ψ comprises 40 group elements.

$\mathcal{P}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}_{\mathsf{pk}}, (X,Y), W, \psi, L, coins_\psi)$: parse $\mathsf{pk}_{\mathsf{GM}}$, $\mathsf{pk}_{\mathsf{OA}}$, pk and ψ as above. Conduct the following steps.

1. Generate commitments (as explained in section 2.3) to the $9n+4=58$ group elements that $\mathsf{cert}_{\mathsf{pk}}$ consists of. The resulting overall commitment $com_{\mathsf{cert}_{\mathsf{pk}}}$ contains 184 group elements.
2. Generate GS commitments to the public key elements $\mathsf{pk} = (X_1, \ldots, X_6)$ and obtain the set $com_{\mathsf{pk}} = \{com_{X_i}\}_{i=1,\ldots,6}$, which consists of 18 group elements.
3. Generate a proof $\pi_{\mathsf{cert}_{\mathsf{pk}}}$ that $com_{\mathsf{cert}_{\mathsf{pk}}}$ is a commitment to a valid certificate for the public key contained in com_{pk}. For each $i=1,\ldots,6$, relations (1)-(3) cost 9 elements to prove (and thus 54 elements altogether). The quadratic equation (4) takes 9 elements and linear ones (5)-(6) both require 3 elements. Finally, (7) is a set of 18 linear equations which demand 54 elements altogether. The whole proof $\pi_{\mathsf{cert}_{\mathsf{pk}}}$ thus takes 123 group elements.
4. For $i=1,\ldots,6$, generate a NIZK proof $\pi_{eq\text{-}key,i}$ that com_{X_i} (which is part of com_{pk}) and ψ_{K_i} are encryptions of the same X_i. If ψ_{K_i} comprises

   $$(V_{i,1}, V_{i,2}, V_{i,5}) = (Y_1^{w_{i,1}}, Y_2^{w_{i,2}}, X_i \cdot g^{w_{i,1}+w_{i,2}})$$

   and com_{X_i} is parsed as $(c_{X_{i1}}, c_{X_{i2}}, c_{X_{i3}}) = (g_1^{\theta_{i1}} \cdot g_{3,1}^{\theta_{i3}},\ g_2^{\theta_{i2}} \cdot g_{3,2}^{\theta_{i3}},\ X_i \cdot g^{\theta_{i1}+\theta_{i2}} \cdot g_{3,3}^{\theta_{i3}})$, where $w_{i,1}, w_{i,2} \in coins_\psi$, $\theta_{i1}, \theta_{i2}, \theta_{i3} \in \mathbb{Z}_p^*$ and $\vec{g_3} = (g_{3,1}, g_{3,2}, g_{3,3})$, this amounts to prove knowledge of values $w_{i,1}, w_{i,2}, \theta_{i1}, \theta_{i2}, \theta_{i3} \in \mathbb{Z}_p^*$ such that

   $$\Big(\frac{V_{i,1}}{c_{X_{i1}}}, \frac{V_{i,2}}{c_{X_{i2}}}, \frac{V_{i,3}}{c_{X_{i3}}}\Big) = \big(Y_1^{w_{i,1}} \cdot g_1^{-\theta_{i1}} \cdot g_{3,1}^{-\theta_{i3}},\ Y_2^{w_{i,2}} \cdot g_2^{-\theta_{i2}} \cdot g_{3,2}^{-\theta_{i3}},\ g^{w_{i,1}+w_{i,2}-\theta_{i1}-\theta_{i2}} \cdot g_{3,3}^{-\theta_{i3}}\big).$$

   Committing to exponents $w_{i,1}, w_{i,2}, \theta_{i1}, \theta_{i2}, \theta_{i3}$ introduces 90 group elements whereas the above relations only require two elements each. Overall, proof elements $\pi_{eq\text{-}key,1}, \ldots, \pi_{eq\text{-}key,6}$ incur 126 elements.
5. Generate a NIZK proof $\pi_{val\text{-}enc}$ that $\psi_{\mathsf{CS}} = (U_1, U_2, U_3, U_4, U_5)$ is a valid CS encryption. This requires to commit to underlying encryption exponents $r, s \in coins_\psi$ and prove that $U_1 = g_1^r$, $U_2 = g_2^s$, $U_3 = g^{r+s}$ (which only takes 3 times 2 elements as base elements are public) and $U_5 = (X_1 X_3^\alpha)^r \cdot (X_2 X_4^\alpha)^s$ (which takes 9 elements since base elements are themselves variables). Including commitments com_r and com_s to exponents r and s, $\pi_{val\text{-}enc}$ demands 21 group elements overall.
6. Generate a NIZK proof $\pi_{\mathcal{R}}$ that the ciphertext ψ_{CS} encrypts a group element $W \in \mathbb{G}$ such that $((X,Y),W) \in \mathcal{R}$. To this end, generate a commitment

   $$com_W = (c_{W,1}, c_{W,2}, c_{W,3}) = (g_1^{\theta_1} \cdot g_{3,1}^{\theta_3},\ g_2^{\theta_2} \cdot g_{3,2}^{\theta_3},\ W \cdot g^{\theta_1+\theta_2} \cdot g_{3,3}^{\theta_3})$$

   and prove that the underlying W is the same as the one for which $U_4 = W \cdot X_5^r \cdot X_6^s$ in ψ_{CS}. In other words, prove knowledge of exponents $r, s, \theta_1, \theta_2, \theta_3$ such that

   $$\Big(\frac{U_1}{c_{W,1}}, \frac{U_2}{c_{W,2}}, \frac{U_4}{c_{W,3}}\Big) = \big(g_1^{r-\theta_1} \cdot g_{3,1}^{-\theta_3},\ g_2^{s-\theta_2} \cdot g_{3,2}^{-\theta_3},\ g^{-\theta_1-\theta_2} \cdot g_{3,3}^{-\theta_3} \cdot X_5^r \cdot X_6^s\big). \quad (8)$$

   Commitments to r, s are already part of $\pi_{val\text{-}enc}$. Committing to $\theta_1, \theta_2, \theta_3$ takes 9 elements. Proving the first two relations of (8) requires 4 elements whereas the third one is quadratic and its proof is 9 elements. Proving the linear pairing-product relation $e(g,W) = e(X,Y)$ in NIZK[6] demands 9 elements. Since $\pi_{\mathcal{R}}$ includes com_W, it entails a total of 34 elements.

[6] It requires to introduce an auxiliary variable $\mathcal{X}$ and prove that $e(g,\mathcal{W}) = e(\mathcal{X},Y)$ and $\mathcal{X} = X$, for variables $\mathcal{W}, \mathcal{X}$ and constants g, X, Y. The two proofs take 3 elements each and 3 elements are needed to commit to $\mathcal{X}$.

The entire proof $\pi_\psi = com_{\mathsf{cert}_{\mathsf{pk}}}||com_{\mathsf{pk}}||\pi_{\mathsf{cert}_{\mathsf{pk}}}||\pi_{eq\text{-}key,1}||\cdots||\pi_{eq\text{-}key,6}||\pi_{val\text{-}enc}||\pi_{\mathcal{R}}$ eventually takes 516 elements.

$\mathcal{V}(\mathsf{param}, \psi, L, \pi_\psi, \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}})$: parse $\mathsf{pk}_{\mathsf{GM}}$, $\mathsf{pk}_{\mathsf{OA}}$, pk, ψ and π_ψ as above. Return 1 if and only if $\mathcal{V}(\mathsf{VK}, \sigma, (\psi_{\mathsf{CS}}||\psi_{\mathsf{K}_1}||\cdots||\psi_{\mathsf{K}_6}||L)) = 1$, all proofs verify and if $\psi_{\mathsf{K}_1}, \ldots, \psi_{\mathsf{K}_6}$ are all valid tag-based encryptions w.r.t. the tag VK.

$\mathsf{DEC}(\mathsf{sk}, \psi, L)$: parse the ciphertext ψ as $\mathsf{VK}||\psi_{\mathsf{CS}}||\psi_{\mathsf{K}_1}||\cdots||\psi_{\mathsf{K}_6}||\sigma$. Return $\perp$ in the event that $\mathcal{V}(\mathsf{VK}, \sigma, (\psi_{\mathsf{CS}}||\psi_{\mathsf{K}_1}||\cdots||\psi_{\mathsf{K}_6}||L)) = 0$. Otherwise, use sk to decrypt (ψ_{CS}, L).

$\mathsf{OPEN}(\mathsf{sk}_{\mathsf{OA}}, \psi, L)$: parse the ciphertext ψ as $\mathsf{VK}||\psi_{\mathsf{CS}}||\psi_{\mathsf{K}_1}||\cdots||\psi_{\mathsf{K}_6}||\sigma$. Return $\perp$ if $\psi_{\mathsf{K}_1}, \ldots, \psi_{\mathsf{K}_6}$ are not all valid TBE ciphertexts w.r.t. the tag VK or if $\mathcal{V}(\mathsf{VK}, \sigma, (\psi_{\mathsf{CS}}||\psi_{\mathsf{K}_1}||\cdots||\psi_{\mathsf{K}_6}||L)) = 0$. Otherwise, decrypt $\psi_{\mathsf{K}_1}, \ldots, \psi_{\mathsf{K}_6}$ using $\mathsf{sk}_{\mathsf{OA}}$ and return the resulting $\mathsf{pk} = (X_1, \ldots, X_6)$.

From an efficiency standpoint, the length of ciphertexts is about 1.25 kB in an implementation using symmetric pairings with a 256-bit group order, which is more compact than in the Paillier-based scheme of [29] where ciphertexts take 2.5 kB using 1024-bit moduli. Moreover, our proofs only require 16.125 kB, which is significantly cheaper than in the original GE scheme [29], where interactive proofs reach a communication cost of 70 kB to achieve a 2^{-50} knowledge error.

References

1. J. H. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption. In *Eurocrypt'02*, LNCS 2332, pages 83–107, 2002.
2. M. Belenkiy, M. Chase, M. Kohlweiss, and A. Lysyanskaya. P-signatures and noninteractive anonymous credentials. In *TCC'08*, LNCS 4948, pages 356–374, 2008.
3. M. Bellare, A. Boldyreva, A. Desai, and D. Pointcheval. Key-privacy in public-key encryption. In *Asiacrypt'01*, LNCS 2248, pages 566–582, 2001.
4. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS'93*, pages 62–73, 1993.
5. A. Boldyreva, M. Fischlin, A. Palacio, and B. Warinschi. A closer look at PKI: Security and efficiency. In *PKC'07*, LNCS 4450, pages 458–475, 2007.
6. D. Boneh and X. Boyen. Short signatures without random oracles. In *Eurocrypt'04*, LNCS 3027, pages 56–73, 2004.
7. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *Crypto'04*, LNCS 3152, pages 41–55, 2004.
8. D. Boneh and M. Franklin. Identity based encryption from the Weil pairing. *SIAM J. of Computing*, 32(3):586–615, 2003. Extended abstract in Crypto'01, LNCS 2139, pages 213–229, 2001.
9. X. Boyen and C. Delerablée. Expressive subgroup signatures. In *SCN'08*, LNCS 5229, pages 185–200, 2008.
10. X. Boyen and B. Waters. Full-domain subgroup hiding and constant-size group signatures. In *PKC'07* , LNCS 4450, pages 1–15, 2007.
11. J. Camenisch, N. Chandran, and V. Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In *Eurocrypt 2009*, LNCS 5479, pages 351–368, 2009.
12. J. Camenisch and A. Lysyanskaya. A Signature Scheme with Efficient Protocols. In *SCN'02*, LNCS 2576, pages 268–289, 2003.
13. J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. In *Crypto'03*, LNCS 2729, pages 126–144, 2003.
14. R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. *Journal of the ACM*, 51(4):557–594, 2004.
15. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *Eurocrypt'04*, LNCS 3027, pages 207–222, 2004.
16. D. Chaum and E. van Heyst. Group signatures. In *Eurocrypt'91*, LNCS 547, pages 257–265, 1991.
17. R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Crypto'98*, LNCS 1462, pages 13–25, 1998.

18. R. Cramer and V. Shoup. Signature schemes based on the strong rsa assumption. In *ACM CCS'99*, pages 46–51, 1999.
19. C. Delerablée and D. Pointcheval. Dynamic fully anonymous short group signatures. In *Viecrypt 2006*, LNCS 4341, pages 193–210, 2006.
20. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Crypto'86*, LNCS 263, pages 186–194, 1986.
21. G. Fuchsbauer and D. Pointcheval. Encrypting Proofs on Pairings and Its Application to Anonymity for Signatures. In *Pairing'09*, LNCS 5671, pages 132–149, 2009.
22. S. Goldwasser and Y. Tauman-Kalai. On the (In)security of the Fiat-Shamir Paradigm In *FOCS'03*, pages 102–115, 2003.
23. J. Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In *Asiacrypt'06*, LNCS 4284, pages 444–459, 2006.
24. J. Groth. Fully anonymous group signatures without random oracles. In *Asiacrypt'07*, LNCS 4833, pages 164–180, 2007.
25. J. Groth. Homomorphic trapdoor commitments to group elements. Cryptology ePrint Archive: Report 2009/007, 2009.
26. J. Groth and S. Lu. A non-interactive shuffle with pairing based verifiability. In *Asiacrypt'07*, LNCS 4833, pages 51–67, 2007.
27. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Eurocrypt'08*, LNCS 4965, pages 415–432, 2008.
28. S. Halevi. A Sufficient Condition for Key-Privacy. Cryptology ePrint Archive: Report 2005/005, 2005.
29. A. Kiayias, Y. Tsiounis, and M. Yung. Group encryption. In *Asiacrypt'07*, LNCS 4833, pages 181–199, 2007.
30. A. Kiayias and M. Yung. Group signatures with efficient concurrent join. In *Eurocrypt'05*, LNCS 3494, pages 198–214, 2005.
31. E. Kiltz. Chosen-ciphertext security from tag-based encryption. In *TCC'06*, LNCS 3876, pages 581–600, 2006.
32. H. Krawczyk and T. Rabin. Chameleon signatures. In *NDSS'00*, 2000.
33. S. Kunz-Jacques and D. Pointcheval. About the security of MTI/C0 and MQV. In *SCN'06*, LNCS 4116, pages 156–172, 2006.
34. M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *STOC'89*, pages 33–43, 1989.
35. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Eurocrypt'99*, *LNCS* 1592, pages 223–238, 1999.
36. B. Qin, Q. Wu, W. Susilo, Y. Mu, Y. Wang. Publicly Verifiable Privacy-Preserving Group Decryption. In *Inscrypt'08*, *LNCS* 5487, pages 72–83, 2008.
37. H. Shacham. A Cramer-Shoup encryption scheme from the linear assumption and from progressively weaker linear variants. Cryptology ePrint Archive: Report 2007/074, 2007.
38. V. Shoup. A proposal for the ISO standard for public-key encryption (version 2.1). manuscript, 2001. `http://shoup.net/`.

A Proof of Theorem 1

The security proof of the certification scheme considers three kinds of forgeries in the attack game.

- Type I forgeries: are such that the fake certificate $\mathsf{cert}^\star_{\mathsf{pk}^\star}$ contains a tuple of elements $(S_1^\star, S_2^\star, S_3^\star)$ that never appeared in outputs of certification queries.
- Type II forgeries: are such that $\mathsf{cert}^\star_{\mathsf{pk}^\star}$ contains a triple $(S_1^\star, S_2^\star, S_3^\star)$ that appeared in the output of some query but $\mathsf{cert}^\star_{\mathsf{pk}^\star}$ also contains commitments $\{(C_{i,1}^\star, C_{i,2}^\star, C_{i,3}^\star)\}_{i=1,\ldots,n}$ that do not match those in the output of that query.
- Type III forgeries: are such that $(S_1^\star, S_2^\star, S_3^\star)$ and $\{(C_{i,1}^\star, C_{i,2}^\star, C_{i,3}^\star)\}_{i=1,\ldots,n}$ are all identical in the fake certificate $\mathsf{cert}^\star_{\mathsf{pk}^\star}$ and in the output of some certification query. On the other hand, the public key $\mathsf{pk}^\star = (X_1^\star, \ldots, X_n^\star)$ is not the one that was certified in that query.

Type I forgeries are easily seen (see lemma 1) to break the HSDH assumption whereas lemma 2 and lemma 3 show that Type II and Type III forgeries give rise to algorithms solving the FlexDH and S2P problems, respectively. □

Lemma 1. *Any Type I forger has advantage at most* $\mathbf{Adv}^{\text{Type-I}}(A) \leq \mathbf{Adv}^{\ell\text{-HSDH}}(\mathcal{B})$*, where* ℓ *is the number of certification queries.*

Proof. The proof is based on ideas from [10]. We outline an algorithm $\mathcal{B}$ that, on input of $\Omega = g^{\omega}$, $u \in \mathbb{G}$ and a set of ℓ triples $(A_i = g^{1/(\omega+c_i)}, B_i = g^{c_i}, C_i = u^{c_i}) \in \mathbb{G}^3$ with $c_1, \ldots, c_\ell \in_R \mathbb{Z}_p^*$, uses a Type I forger to find a triple $(g^{1/(\omega+c)}, g^c, u^c)$ such that $c \neq c_i$ for $i = 1, \ldots, \ell$. To generate the public key PK, $\mathcal{B}$ chooses $\beta_0 \xleftarrow{\$} \mathbb{Z}_p^*$, $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$ and sets $u_0 = g^{\beta_0}$, $A = e(g,g)^{\alpha}$. It also defines $\{(u_{i,1} = g^{\beta_{i,1}}, u_{i,2} = g^{\beta_{i,2}}, u_{i,3} = g^{\beta_{i,2}})\}_{i=1,\ldots,n}$ using random triples $(\beta_{i,1}, \beta_{i,2}, \beta_{i,3}) \xleftarrow{\$} (\mathbb{Z}_p^*)^3$ for $i = 1, \ldots, n$ whereas $\vec{f}_1, \vec{f}_2, \vec{f}_3$ are defined by $f = g^{\theta_0}, f_1 = g^{\theta_1}, f_2 = g^{\theta_2}, f_{3,1} = g^{\theta_1 \xi_1}, f_{3,2} = g^{\theta_2 \xi_2}$ and $f_{3,3} = g^{\theta_0(\xi_1+\xi_2+\xi_3)}$ for random chosen $\theta_0, \theta_1, \theta_2, \xi_1, \xi_2, \xi_3 \xleftarrow{\$} \mathbb{Z}_p^*$.

To answer certification queries involving public keys $\mathsf{pk} = (X_1, \ldots, X_n)$, $\mathcal{B}$ first computes n commitments to $1_{\mathbb{G}}$. For $i = 1, \ldots, n$, it randomly picks $\phi_{i,1}, \phi_{i,2}, \phi_{i,3} \xleftarrow{\$} \mathbb{Z}_p^*$ and computes a commitment $\vec{C}_i = (f_1^{\phi_{i,1}} \cdot f_{3,1}^{\phi_{i,3}}, f_2^{\phi_{i,2}} \cdot f_{3,2}^{\phi_{i,3}}, f^{\phi_{i,1}+\phi_{i,2}} \cdot f_{3,3}^{\phi_{i,3}})$, which equals $(g^{\eta_{i,1}}, g^{\eta_{i,2}}, g^{\eta_{i,3}})$ where $\eta_{i,1} = \theta_1(\phi_{i,1} + \xi_1 \phi_{i,3})$, $\eta_{i,2} = \theta_2(\phi_{i,2} + \xi_2 \phi_{i,3})$ and $\eta_{i,3} = \theta_0((\phi_{i,1} + \phi_{i,2}) + (\xi_1 + \xi_2 + \xi_3)\phi_{i,3})$ are all known to $\mathcal{B}$. Certificate parts (S_1, S_2, S_3) are generated as (A_k^{α}, B_k, C_k) using the next available triple (A_k, B_k, C_k) (with $k \in \{1, \ldots, \ell\}$). Finally, as for remaining certificate elements $S_4 = (u_0 \cdot \prod_{i=1}^n (C_{i,1}^{\beta_{i,1}} \cdot C_{i,2}^{\beta_{i,2}} \cdot \mathcal{C}_{i,3}^{\beta_{i,3}}))^{c_k}$ and $S_5 = \{(C_{i,1}^{c_k}, C_{i,2}^{c_k}, C_{i,3}^{c_k})\}_{i=1,\ldots,n}$, they are calculated as $S_4 = B_k^{\beta_0 + \sum_{i=1}^n \sum_{j=1}^3 \beta_{i,j}\eta_{i,j}}$ and $S_5 = \{(B_k^{\eta_{i,1}}, B_k^{\eta_{i,2}}, B_k^{\eta_{i,3}}\}_{i=1,\ldots,n}$, respectively. To complete the generation of $\mathsf{cert}_{\mathsf{pk}}$, $\mathcal{B}$ then trapdoor opens $\vec{C}_1, \ldots, \vec{C}_n$ to $(X_1, \ldots, X_n)$ using the trapdoor (ξ_1, ξ_2, ξ_3). More precisely, for $i = 1, \ldots, n$, it computes the de-commitments

$$(D'_{i,1}, D'_{i,2}, D'_{i,3}) = \big(f^{\phi_{i,1}} \cdot (X_i)^{\xi_1/\xi_3}, f^{\phi_{i,2}} \cdot (X_i)^{\xi_2/\xi_3}, f^{\phi_{i,3}} \cdot (1/X_i)^{1/\xi_3}\big).$$

The game ends with $\mathcal{A}$ outputting a pair $(\mathsf{pk}^\star, \mathsf{cert}^\star_{\mathsf{pk}^\star})$ such that $(S_1^\star, S_2^\star, S_3^\star)$ never appeared within outputs of certification queries. Hence, $(S_1^{\star 1/\alpha}, S_2^\star, S_3^\star)$ must solve the HSDH problem. □

Lemma 2. *Any Type II forger* $\mathcal{A}$ *making* ℓ *certification queries has no better advantage than* $\mathbf{Adv}^{\text{Type-II}}(A) \leq \ell \cdot (1 - \frac{1}{p}) \cdot \mathbf{Adv}^{\text{FlexDH}}(\mathcal{B})$.

Proof. We show how a Type II forger implies an algorithm $\mathcal{B}$ that finds a non-trivial triple (C, C^a, C^{ab}) on input of $(g, g_a = g^a, g_b = g^b)$. To generate PK, $\mathcal{B}$ chooses $\omega \xleftarrow{\$} \mathbb{Z}_p^*$, $\alpha_a, \alpha_u \xleftarrow{\$} \mathbb{Z}_p^*$ and sets $\Omega = g^{\omega}$, $A = e(g, (g_a \cdot g^{\omega})^{\alpha_a}))$ (so that $\alpha = \log_{e(g,g)}(A)$ is implicitly set as $\alpha = (a+\omega)\alpha_a)$ and $u = g^{\alpha_u}$. The commitment key $(\vec{f}_1, \vec{f}_2, \vec{f}_3)$ is obtained by choosing $\theta, \theta_1, \theta_2, \xi_1, \xi_2, \xi_3 \xleftarrow{\$} \mathbb{Z}_p^*$ and setting $f = g^{\theta}, f_1 = g^{\theta_1}, f_2 = g^{\theta_2}, f_{3,1} = g^{\theta_1 \xi_1}, f_{3,2} = g^{\theta_2 \xi_2}$ and $f_{3,3} = g^{\theta(\xi_1+\xi_2+\xi_3)}$.

In the setup phase, $\mathcal{B}$ also computes a set of n commitments to $1_{\mathbb{G}}$, say $\vec{C}_i^\dagger = (g^{\eta_{i,1}^\dagger}, g^{\eta_{i,2}^\dagger}, g^{\eta_{i,3}^\dagger})$ for $i = 1, \ldots, n$, and obtains them by drawing random exponents $\phi_{i,1}^\dagger, \phi_{i,2}^\dagger, \phi_{i,3}^\dagger \xleftarrow{\$} \mathbb{Z}_p^*$ and computing $\eta_{i,1}^\dagger = \theta_1(\phi_{i,1}^\dagger + \xi_1 \phi_{i,3}^\dagger)$, $\eta_{i,2}^\dagger = \theta_2(\phi_{i,2}^\dagger + \xi_2 \phi_{i,3}^\dagger)$ as well as $\eta_{i,3}^\dagger = \theta((\phi_{i,1}^\dagger + \phi_{i,2}^\dagger) + (\xi_1 + \xi_2 + \xi_3)\phi_{i,3}^\dagger)$. It also retains $\phi_{i,1}^\dagger, \phi_{i,2}^\dagger, \phi_{i,3}^\dagger$ for later use. Next, $\mathcal{B}$ picks $\rho \xleftarrow{\$} \mathbb{Z}_p^*$ and two sets of n triples $(\rho_{i,1}, \rho_{i,2}, \rho_{i,3}) \xleftarrow{\$} (\mathbb{Z}_p^*)^3$, $(\gamma_{i,1}, \gamma_{i,2}, \gamma_{i,3}) \xleftarrow{\$} (\mathbb{Z}_p^*)^3$ and defines $u_{i,1} = g^{\rho_{i,1}} \cdot g_b^{\gamma_{i,1}}$, $u_{i,2} = g^{\rho_{i,2}} \cdot g_b^{\gamma_{i,2}}$, $u_{i,3} = g^{\rho_{i,3}} \cdot g_b^{\gamma_{i,3}}$, for $i = 1, \ldots, n$, and $u_0 = g^{\rho} \cdot g_b^{\gamma}$ with $\gamma = -\sum_{i=1}^n (\gamma_{i,1}\eta_{i,1}^\dagger + \gamma_{i,2}\eta_{i,2}^\dagger + \gamma_{i,3}\eta_{i,3}^\dagger)$. This implicitly defines private key elements to be $\beta_{i,1} = \rho_{i,1} + b\gamma_{i,1}$, $\beta_{i,2} = \rho_{i,2} + b\gamma_{i,2}$ and $\beta_{i,3} = \rho_{i,3} + b\gamma_{i,3}$. At the outset of the game, $\mathcal{B}$ also chooses $\ell^\star \xleftarrow{\$} \{1, \ldots, \ell\}$.

When the certification of a public key $\mathsf{pk} = (X_1, \ldots, X_n)$ is queried, the query's treatment depends on the index $k \in \{1, \ldots, \ell\}$ of the query.

- If $k \neq \ell^\star$, $\mathcal{B}$ computes n commitments to $1_{\mathbb{G}}$ (say $\vec{C}_i = (g^{\eta_{i,1}}, g^{\eta_{i,2}}, g^{\eta_{i,3}})$ for $i = 1, \ldots, n$) as in the proof of lemma 1. It generates $\mathsf{cert}_{\mathsf{pk}}$ by setting $S_1 = (g_a \cdot g^{\omega})^{\alpha_a/(\omega+c_k)}$, $S_2 = g^{c_k}$, $S_3 = u^{c_k}$ for a random $c_k \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$. Since it knows $\prod_{i=1}^n (C_{i,1}^{\beta_{i,1}} \cdot C_{i,2}^{\beta_{i,2}} \cdot \mathcal{C}_{i,3}^{\beta_{i,3}}) = \prod_{i=1}^n (u_{i,1}^{\eta_{i,1}} \cdot u_{i,2}^{\eta_{i,2}} \cdot u_{i,3}^{\eta_{i,3}})$ (thanks to $(\eta_{i,1}, \eta_{i,2}, \eta_{i,3})$), it can also compute $S_4 = (u_0 \cdot \prod_{i=1}^n (C_{i,1}^{\beta_{i,1}} \cdot C_{i,2}^{\beta_{i,2}} \cdot \mathcal{C}_{i,3}^{\beta_{i,3}}))^{c_k}$. Finally, $S_5 = \{(C_{i,1}^{c_k}, C_{i,2}^{c_k}, C_{i,3}^{c_k})\}_{i=1,\ldots,n}$ is also computable from c_k and $\mathcal{B}$ uses ξ_1, ξ_2, ξ_3 to trapdoor open $\vec{C}_i$ to X_i and obtain de-commitments $(D'_{i,1}, D'_{i,2}, D'_{i,3})$, for $i = 1, \ldots, n$, as in lemma 1.
- If $k = \ell^\star$, $\mathcal{B}$ implicitly defines $c_{\ell^\star} = a$ and sets $S_1 = g^{\alpha_a}$, $S_2 = g_a$, $S_3 = g_a^{\alpha_u}$. Thanks to $\vec{C}_i^\dagger = (C_{i,1}^\dagger, C_{i,2}^\dagger, C_{i,3}^\dagger) = (g^{\eta_{i,1}^\dagger}, g^{\eta_{i,2}^\dagger}, g^{\eta_{i,3}^\dagger})$ that were chosen in the setup phase, $\mathcal{B}$ can compute
$$S_4 = \big(u_0 \cdot \prod_{i=1}^n ({C_{i,1}^\dagger}^{\beta_{i,1}} \cdot {C_{i,2}^\dagger}^{\beta_{i,2}} \cdot {C_{i,3}^\dagger}^{\beta_{i,3}})\big)^a = g_a^{\rho + \sum_{i=1}^n (\rho_{i,1}\eta_{i,1}^\dagger + \rho_{i,2}\eta_{i,2}^\dagger + \rho_{i,3}\eta_{i,3}^\dagger)}.$$
Finally, $\mathcal{B}$ obtains $S_5 = \{({C_{i,1}^\dagger}^a, {C_{i,2}^\dagger}^a, {C_{i,3}^\dagger}^a) = (g_a^{\eta_{i,1}^\dagger}, g_a^{\eta_{i,2}^\dagger}, g_a^{\eta_{i,3}^\dagger})\}_{i=1,\ldots,n}$. For $i = 1, \ldots, n$, it trapdoor opens $\vec{C}_i^\dagger$ to public key elements X_i using the trapdoor (ξ_1, ξ_2, ξ_3) and the de-commitment $(f^{\phi_{i,1}^\dagger}, f^{\phi_{i,2}^\dagger}, f^{\phi_{i,3}^\dagger})$ that was associated with the commitment to $1_{\mathbb{G}}$. The resulting de-commitment $(D'_{i,1}, D'_{i,2}, D'_{i,3})$ and $\vec{C}_i^\dagger = (C_{i,1}^\dagger, C_{i,2}^\dagger, C_{i,3}^\dagger)$ are included in $\mathsf{cert}_{\mathsf{pk}}$.

Finally, $\mathcal{A}$ outputs a pair $(\mathsf{pk}^\star, \mathsf{cert}^\star_{\mathsf{pk}^\star})$ such that $(S_1^\star, S_2^\star, S_3^\star)$ appeared in the output of some certification query but $\mathsf{cert}^\star_{\mathsf{pk}^\star}$ comprises commitments $\{(C_{i,1}^\star, C_{i,2}^\star, C_{i,3}^\star)\}_{i=1,\ldots,n}$ that do not match those returned in that specific query. With probability $1/\ell$, this query happens to be the $\ell^{\star\text{th}}$ one (and $\mathcal{B}$ fails if this is not the case), so that $(S_1^\star, S_2^\star, S_3^\star) = (g^{\alpha_a}, g_a, g_a^{\alpha_u})$. Then, we must have
$$S_4^\star = \big(u_0 \cdot \prod_{i=1}^n ({C_{i,1}^\star}^{\beta_{i,1}} \cdot {C_{i,2}^\star}^{\beta_{i,1}} \cdot {C_{i,3}^\star}^{\beta_{i,3}})\big)^a, \qquad S_5^\star = \{({C_{i,1}^\star}^a, {C_{i,2}^\star}^a, {C_{i,3}^\star}^a)\}_{i=1,\ldots,n},$$
where $\{(C_{i,1}^\star, C_{i,2}^\star, C_{i,3}^\star)\}_{i=1,\ldots,n} \neq \{(C_{i,1}^\dagger, C_{i,2}^\dagger, C_{i,3}^\dagger)\}_{i=1,\ldots,n}$, in such a way that dividing out the value $S_4 = g_a^{\rho + \sum_{i=1}^n (\rho_{i,1}\eta_{i,1}^\dagger + \rho_{i,2}\eta_{i,2}^\dagger + \rho_{i,3}\eta_{i,3}^\dagger)}$ from $S_4^\star$ yields
$$T = \big(\prod_{i=1}^n (C_{i,1}^\star / C_{i,1}^\dagger)^{\rho_{i,1} + b\gamma_{i,1}} \cdot (C_{i,2}^\star / C_{i,2}^\dagger)^{\rho_{i,2} + b\gamma_{i,2}} \cdot (C_{i,3}^\star / C_{i,3}^\dagger)^{\rho_{i,3} + b\gamma_{i,3}}\big)^a$$
whereas the component-wise quotient of $S_5 = \{(g_a^{\eta_{i,1}^\dagger}, g_a^{\eta_{i,2}^\dagger}, g_a^{\eta_{i,3}^\dagger})\}_{i=1,\ldots,n}$ and S_5^* reveals a triple $\{(Z_{i,1}, Z_{i,2}, Z_{i,3}) = \big((C_{i,1}^\star / C_{i,1}^\dagger)^a, (C_{i,2}^\star / C_{i,2}^\dagger)^a, (C_{i,3}^\star / C_{i,3}^\dagger)^a\big)\}_{i=1,\ldots,n}$. Hence, $\mathcal{B}$ extracts
$$R_3 = \Big(\prod_{i=1}^n (\frac{C_{i,1}^\star}{C_{i,1}^\dagger})^{\gamma_{i,1}} \cdot (\frac{C_{i,2}^\star}{C_{i,2}^\dagger})^{\gamma_{i,2}} \cdot (\frac{C_{i,3}^\star}{C_{i,3}^\dagger})^{\gamma_{i,3}}\Big)^{ab} = T / \prod_{i=1}^n Z_{i,1}^{\rho_{i,1}} \cdot Z_{i,2}^{\rho_{i,2}} \cdot Z_{i,3}^{\rho_{i,3}},$$
$$R_2 = \prod_{i=1}^n Z_{i,1}^{\gamma_{i,1}} \cdot Z_{i,2}^{\gamma_{i,2}} \cdot Z_{i,3}^{\gamma_{i,3}} = \Big(\prod_{i=1}^n (\frac{C_{i,1}^\star}{C_{i,1}^\dagger})^{\gamma_{i,1}} \cdot (\frac{C_{i,2}^\star}{C_{i,2}^\dagger})^{\gamma_{i,2}} \cdot (\frac{C_{i,3}^\star}{C_{i,3}^\dagger})^{\gamma_{i,3}}\Big)^a$$
$$R_1 = \prod_{i=1}^n (\frac{C_{i,1}^\star}{C_{i,1}^\dagger})^{\gamma_{i,1}} \cdot (\frac{C_{i,2}^\star}{C_{i,2}^\dagger})^{\gamma_{i,2}} \cdot (\frac{C_{i,3}^\star}{C_{i,3}^\dagger})^{\gamma_{i,3}}$$
which must form a non-trivial triple (R_1, R_1^a, R_1^{ab}) with overwhelming probability. Indeed, since $\gamma_{i,1}, \gamma_{i,2}, \gamma_{i,3}$ are (information theoretically) independent of $\mathcal{A}$'s view, we can only have $R_1 = 1_{\mathbb{G}}$ by pure chance (with probability $1/p$). □

Lemma 3. *Any Type III forger has advantage at most* $\mathbf{Adv}^{\text{Type-III}}(A) \leq \mathbf{Adv}^{\text{S2P}}(\mathcal{B})$.

Proof. From a Type III adversary $\mathcal{A}$, it is simple to break the binding property of the commitment scheme in section 3.1. Consider an algorithm $\mathcal{B}$ which is given a commitment key $(\vec{f_1}, \vec{f_2}, \vec{f_3})$ and prepares the rest of the public key according to the specification of the scheme in such a way that it can perfectly answer all certification queries.

At the end of the game, $\mathcal{A}$ outputs a pair $\mathsf{pk}^\star$ and $\mathsf{cert}^\star_{\mathsf{pk}^\star}$ such that $\mathsf{cert}_{\mathsf{pk}^\star}$ contains $(S_1^\star, S_2^\star, S_2^\star)$ and commitments $(C^\star_{i,1}, C^\star_{i,2}, C^\star_{i,3})$ that were both contained in the output of some certification query. On the other hand, the public key $\mathsf{pk}^* = (X_1^\star, \ldots, X_n^\star)$ must be different from the one $(X_1, \ldots, X_n)$ that was certified at that query. This necessarily provides $\mathcal{B}$ with two distinct openings $(X_i, (D_{1,i}, D_{2,i}, D_{3,i}))$, $(X_i^\star, (D^\star_{1,i}, D^\star_{2,i}, D^\star_{3,i}))$ (since $X_i \neq X_i^\star$ for at least one index $i \in \{1, \ldots, n\}$) of some commitment $(C^\star_{i,1}, C^\star_{i,2}, C^\star_{i,3})$, which violates the S2P assumption. □

B Security Proofs for the Group Encryption Scheme

Correctness is straightforward and we focus on anonymity, message security and soundness.

Theorem 2. *The* GE *scheme satisfies anonymity assuming that* Σ *is strongly unforgeable, that* H *is collision-resistant and that the DLIN assumption holds in* $\mathbb{G}$.

Proof. We consider a sequence of games where the first game is the real experiment of definition 6 while the adversary $\mathcal{A}$ is essentially a key privacy attacker against the linear Cramer-Shoup system in the last game. In Game i, we call W_i the event that $\mathcal{A}$ wins.

Game 1: the challenger $\mathcal{B}$ generates param that includes a reference string $\mathbf{g}$ containing $\vec{g_1}$, $\vec{g_2}$ and $\vec{g_3} = \vec{g_1}^{\xi_1} \odot \vec{g_2}^{\xi_2}$, with $\xi_1, \xi_2 \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$. The public key $\mathsf{pk}_{\mathsf{OA}} = (Y_1, Y_2, Y_3, Y_4)$ is given to $\mathcal{A}$ who generates $\mathsf{pk}_{\mathsf{GM}}$ on her own. By invoking the USER oracle, she certifies two distinct receivers' public keys $\mathsf{pk}_0 = (X_1^0, \ldots, X_6^0)$, $\mathsf{pk}_1 = (X_1^1, \ldots, X_6^1)$ chosen by $\mathcal{B}$ and makes a number of opening queries and decryption queries, which $\mathcal{B}$ handles using $\mathsf{sk}_{\mathsf{OA}}$ and sk_0, sk_1, respectively. At some point, she outputs $((X, Y), W, L, \mathsf{pk}_{\mathcal{R}})$ such that $((X, Y), W) \in \mathcal{R}$ and obtains, as a challenge, a group encryption $\psi^\star = \mathsf{VK}^\star || \psi^\star_{\mathsf{CS}} || \psi^\star_{\mathsf{K}_1} || \cdots || \psi^\star_{\mathsf{K}_6} || \sigma^\star$ of W under pk_b, for some bit $b \in \{0, 1\}$ of $\mathcal{B}$'s choice. Then, she obtains proofs $\pi^\star_{\psi^\star}$ for $\psi^\star$ and makes new opening and decryption queries under the obvious restrictions. She finally outputs b' and we call W_1 the event that $b' = b$.

Game 2: is as game 1 but $\mathcal{B}$ aborts in the event F_2 that $\mathcal{A}$ queries the opening of a ciphertext $\psi = \mathsf{VK} || \psi_{\mathsf{CS}} || \psi_{\mathsf{K}_1} || \cdots || \psi_{\mathsf{K}_6} || \sigma$ such that $\mathsf{VK} = \mathsf{VK}^\star$ and σ is valid (we may assume that $\mathsf{VK}^\star$ is generated at the outset of the game). If F_2 occurs, $\mathcal{A}$ is necessarily able to break the strong security of Σ (even if the query occurs before the challenge phase, $\mathcal{A}$ has forged a signature without seeing any signature) and $|\Pr[W_2] - \Pr[W_1]| \leq \Pr[F_2] \in \mathsf{negl}(\lambda)$ if Σ is strongly unforgeable.

Game 3: we modify the generation of the common reference string $\mathbf{g} = (\vec{g_1}, \vec{g_2}, \vec{g_3})$ in param and choose the vector $\vec{g_3}$ as $\vec{g_3} = \vec{g_1}^{\xi_1} \odot \vec{g_2}^{\xi_2} \odot (1, 1, g)^{-1}$ (instead of $\vec{g_3} = \vec{g_1}^{\xi_1} \odot \vec{g_2}^{\xi_2}$). Under the DLIN assumption, this change is not noticeable to $\mathcal{A}$ and $|\Pr[W_3] - \Pr[W_2]| \in \mathsf{negl}(\lambda)$.

Game 4: we change the generation of proofs $\pi^\star_{\psi^\star}$ and use the trapdoor of the CRS (*i.e.*, values ξ_1 and ξ_2 such that $\vec{g_3} = \vec{g_1}^{\xi_1} \odot \vec{g_2}^{\xi_2} \odot (1, 1, g)^{-1}$ and commitments to exponents are thus generated using $\vec{\varphi} = \vec{g_1}^{\xi_1} \odot \vec{g_2}^{\xi_2}$) instead of some of the actual witnesses. Namely, $\pi^\star_{eq\text{-}key,1}, \ldots, \pi^\star_{eq\text{-}key,6}$ as well as $\pi^\star_{val\text{-}enc}$ and $\pi^\star_{\mathcal{R}}$ are simulated without using encryption exponents $\{w^\star_{i,1}, w^\star_{i,2}\}_{i=1,\ldots,6}$ (that are used to encrypt $\psi^\star_{\mathsf{K}_i}$) and $r^\star, s^\star$ (that are used to compute $\psi^\star_{\mathsf{CS}}$). Commitments $com^\star_{pk_b}$ and $com^\star_W$ are still generated using pk_b and W but commitments to $w^\star_{i,1}, w^\star_{i,2}$ and $r^\star, s^\star$ are replaced by commitments to 0. Yet, the trapdoor ξ_1, ξ_2 allows generating proofs that have the same distribution as real proofs (e.g., [11, Section 4.4] shows how to simulate linear multi-exponentiation equations whereas quadratic ones, such as the third relation of (8), are also simulatable without $r^\star, s^\star$) and it comes that $\Pr[W_4] = \Pr[W_3]$.

Game 5: we modify the generation of the challenge ciphertext $\psi^\star$ and let $\psi^\star_{\mathsf{K}_1}, \ldots, \psi^\star_{\mathsf{K}_6}$ be encryptions of random group elements instead of $X_1^b, \ldots, X_6^b$. Since exponents $\{w^\star_{i,1}, w^\star_{i,2}\}_{i=1,\ldots,6}$ are no longer used in Game 4, any significant change in the distribution of $\mathcal{A}$'s output would give rise to a selective-tag weak CCA2 attacker[7] against the tag-based encryption (recall that opening queries do not involve $\mathsf{VK}^\star$ unless the rejection rule of Game 2 applies). According to theorem 5.1 in [31], we have $|\Pr[W_5] - \Pr[W_4]| \in \mathsf{negl}(\lambda)$ if DLIN holds.

In Game 5, $\mathcal{A}$ is essentially playing a CCA2 anonymity attack against the linear Cramer-Shoup encryption scheme. Indeed, elements $\psi^\star_{\mathsf{K}_i}$ do not depend on b and, since proofs are given in the WI setting, they reveal no information on underlying witnesses (in particular, $com^\star_{\mathsf{pk}_b}$ and $com^\star_{\mathsf{cert}_{\mathsf{pk}_b}}$ are perfectly hiding commitments). As for the key privacy of the linear Cramer-Shoup encryption scheme, re-proving theorem 6 in [3] using DLIN in place of DDH is just an exercise and we eventually obtain $|\Pr[W_5] - 1/2| \in \mathsf{negl}(\lambda)$ if DLIN holds and if H is collision-resistant. □

Theorem 3. *The GE scheme satisfies message security assuming that Σ is strongly unforgeable, that H is a collision-resistant hash function and that the DLIN assumption holds in $\mathbb{G}$.*

Proof. We use a sequence of games. The first one mirrors the experiment of definition 5 where the challenger's bit b is 1 and the adversary obtains a encryption of the witness W and real proofs when invoking the $\mathsf{PROVE}(.)$ oracle. In the last game, the adversary $\mathcal{A}$ obtains an encryption of a random plaintext and proofs are simulated using a fake CRS (constructing a simulator for $\mathsf{PROVE}(.)$ using a simulated CRS is part of the security analysis as stressed in [29]). In Game i, W_i denotes the event that $\mathcal{A}$ outputs $b' = 1$.

Game 1: the challenger $\mathcal{B}$ provides $\mathcal{A}$ with common public parameters param that include a real CRS $\mathbf{g}$ containing $(\vec{g_1}, \vec{g_2}, \vec{g_3} = \vec{g_1}^{\xi_1} \odot \vec{g_2}^{\xi_2})$, with $\xi_1, \xi_2 \xleftarrow{\$} \mathbb{Z}_p^*$. The adversary generates public keys $\mathsf{pk}_{\mathsf{OA}}$ and $\mathsf{pk}_{\mathsf{GM}}$ on her own. The challenger and $\mathcal{A}$ run an execution of JOIN where $\mathcal{A}$ certifies the public key $\mathsf{pk} = (X_1, \ldots, X_6)$ of a honest receiver chosen by $\mathcal{B}$. Then, $\mathcal{A}$ makes a number of decryption queries that $\mathcal{B}$ handles using the private key sk that matches pk. At some point, $\mathcal{A}$ outputs $((X,Y), W, L, \mathsf{pk}_{\mathcal{R}})$ such that $((X,Y), W) \in \mathcal{R}$ and obtains in return a group encryption $\psi^\star = \mathsf{VK}^\star || \psi^\star_{\mathsf{CS}} || \psi^\star_{\mathsf{K}_1} || \cdots || \psi^\star_{\mathsf{K}_6} || \sigma^\star$ of W under pk and L. Then, she obtains polynomially many proofs $\pi^\star_{\psi^\star}$ for $\psi^\star$ and makes new decryption queries under the obvious restrictions. She finally outputs b' and we call W_1 the event that $b' = 1$.

Game 2: we modify the generation of the common reference string $\mathbf{g} = (\vec{g_1}, \vec{g_2}, \vec{g_3})$ in param and choose $\vec{g_3} = \vec{g_1}^{\xi_1} \odot \vec{g_2}^{\xi_2} \odot (1,1,g)^{-1}$ (instead of $\vec{g_3} = \vec{g_1}^{\xi_1} \odot \vec{g_2}^{\xi_2}$). Under the DLIN assumption, this change is not noticeable to $\mathcal{A}$ and $|\Pr[W_2] - \Pr[W_1]| \in \mathsf{negl}(\lambda)$.

Game 3: we modify the $\mathsf{DEC}(.)$ oracle and let $\mathcal{B}$ reject any ciphertext of the form $\psi = \mathsf{VK} || \cdots || \sigma$ such that $\mathsf{VK} = \mathsf{VK}^\star$ ($\mathsf{VK}^\star$ can be generated at the outset of the game). Let F_3 be the event that this rule causes $\mathcal{B}$ to reject a ciphertext that would not have been rejected in Game 2. As in the

[7] Selective-tag weak CCA2 security is defined [31] via a game where the adversary $\mathcal{A}$ chooses a tag $t^\star$ and then obtains a public key and access to a decryption oracle which she can query for any ciphertext-tag pair (C, t) such that $t \neq t^\star$. At the challenge phase, she chooses plaintexts m_0, m_1 and receives a ciphertext $C^\star$ encrypting m_b (under the tag $t^\star$) for some bit $b \xleftarrow{\$} \{0,1\}$ that $\mathcal{A}$ eventually aims to guess after further decryption queries.

proof of theorem 2, we have $|\Pr[W_3] - \Pr[W_2]| \leq \Pr[F_3] \in \mathsf{negl}(\lambda)$ if Σ is strongly unforgeable.

Game 4: we change the generation of proofs $\pi^\star_{\psi^\star}$ and use the trapdoor of the CRS instead of witnesses W and $coins_{\psi^\star} = \{(r^\star, s^\star), \{(w^\star_{i,1}, w^\star_{i,2})\}_{i=1,\ldots,6}\}$. More precisely, $\{\pi^\star_{eq\text{-}key,i}\}_{i=1,\ldots,6}$ (which prove that com_{X_i} and ψ_{K_i} hide the same X_i), as well as $\pi^\star_{val\text{-}enc}$ and $\pi^\star_{\mathcal{R}}$ (*i.e.*, the proofs that $\psi^\star_{\mathsf{CS}}$ is a valid ciphertext and that $\psi^\star_{\mathsf{CS}}$ and com_W contain the same W) are simulated without using encryption exponents $\{w^\star_{i,1}, w^\star_{i,2}\}_{i=1,\ldots,6}$ and $r^\star, s^\star$ and commitments to the latter values are replaced by commitments to 0. Also, the part of $\pi^\star_{\mathcal{R}}$ that proves relation $e(g, W) = e(X, Y)$ (and thus $((X, Y), W) \in \mathcal{R}$) is simulated in NIZK[8] by setting com_W as a commitment to $1_{\mathbb{G}}$. As in the proof of theorem 2, the trapdoor ξ_1, ξ_2 allows generating simulated proofs that are perfectly indistinguishable from real proofs, so that $\Pr[W_4] = \Pr[W_3]$.

Game 5: in the calculation of $\psi^\star$, we set $\psi^\star_{\mathsf{CS}}$ as an encryption of a random group element. Since $r^\star, s^\star$ are not used in Game 4, any significant change in $\mathcal{A}$'s behavior would imply a CCA2 attacker (in the modeling of CCA2-security for labeled cryptosystems [38]) against the linear Cramer-Shoup scheme (recall that decryption queries do not involve $\mathsf{VK}^\star$ unless the rejection rule of Game 3 applies, which prevents $\mathcal{A}$ from mauling $\psi^\star_{\mathsf{K}_i}$ while keeping the same $(\psi^\star_{\mathsf{CS}}, \mathsf{VK}^\star, L)$). The result of [37] implies that $|\Pr[W_5] - \Pr[W_4]| \in \mathsf{negl}(\lambda)$ if DLIN holds and H is a collision-resistant hash function.

Game 6: we change again the $\mathsf{DEC}(.)$ oracle and do not apply the rejection rule of Game 3 anymore. If Σ is strongly unforgeable, we must have $|\Pr[W_6] - \Pr[W_5]| \in \mathsf{negl}(\lambda)$.

We see that, from Game 4 onwards, the oracle $\mathsf{PROVE}(.)$ does not use witnesses $W, coins_{\psi^\star}$ any longer. Game 6 is thus the experiment of definition 5 where the challenger's bit b is 0. Putting the above altogether, we find $|\Pr[W_6] - \Pr[W_1]| \in \mathsf{negl}(\lambda)$, which establishes the result. □

Soundness directly follows from the security of the certification system. From a soundness adversary, the simulator interacts with a challenger for the certification security game and generates the CRS $\mathbf{g}$ for the perfect soundness setting (which precludes the generation of valid proofs for ill-formed ciphertexts). Then, soundness can only be broken by attacking the certification scheme.

[8] In addition to the variable $\mathcal{W}$, the latter proof introduces an auxiliary variable $\mathcal{X}$ and provides evidence that $e(g, \mathcal{W}) = e(\mathcal{X}, Y)$ and $\mathcal{X} = X$, for constants g, X, Y. The NIZK simulator can use witnesses $\mathcal{X} = \mathcal{W} = 1_{\mathbb{G}}$ to prove the relation $e(g, \mathcal{W}) = e(\mathcal{X}, Y)$ and simulate a proof for the second relation thanks to the trapdoor of the fake CRS.

Scalable Group Signatures with Revocation

Benoît Libert[1] *, Thomas Peters[1] **, and Moti Yung[2]

[1]Université catholique de Louvain, ICTEAM Institute (Belgium)
[2] Google Inc. and Columbia University (USA)

Abstract. Group signatures are a central cryptographic primitive, simultaneously supporting accountability and anonymity. They allow users to anonymously sign messages on behalf of a group they are members of. The recent years saw the appearance of several constructions with security proofs in the standard model (*i.e.*, without appealing to the random oracle heuristic). For a digital signature scheme to be adopted, an efficient revocation scheme (as in regular PKI) is absolutely necessary. Despite over a decade of extensive research, membership revocation remains a non-trivial problem in group signatures: all existing solutions are not truly scalable due to either high overhead (e.g., large group public key size), or limiting operational requirement (the need for all users to follow the system's entire history). In the standard model, the situation is even worse as many existing solutions are not readily adaptable. To fill this gap and tackle this challenge, we describe a new revocation approach based, perhaps somewhat unexpectedly, on the Naor-Naor-Lotspiech framework which was introduced for a different problem (namely, that of broadcast encryption). Our mechanism yields efficient and scalable revocable group signatures in the standard model. In particular, the size of signatures and the verification cost are independent of the number of revocations and the maximal cardinality N of the group while other complexities are at most polylogarithmic in N. Moreover, the schemes are history-independent: unrevoked group members do not have to update their keys when a revocation occurs.

Keywords. Group signatures, revocation, standard model, efficiency.

1 Introduction

As suggested by Chaum and van Heyst in 1991 [32], *group signatures* allow members of a group to anonymously sign messages on behalf of a population group members managed by a group authority. Using some trapdoor information, a tracing authority must be able to "open" signatures and identify the signer. A complex problem in group signatures is the revocation of members whose signing capability should be disabled (either because they misbehaved or they intentionally leave the group).

1.1 Related Work

Group signatures without revocation. The first provably coalition-resistant scalable group signature was described by Ateniese, Camenisch, Joye and Tsudik in 2000 [7]. At that time, the security of group signatures was not totally understood and proper security definitions were given later on by Bellare, Micciancio and Warinschi [9] (BMW) whose model captures all the requirements of group signatures in three properties. In (a relaxation of) this model, Boneh, Boyen and Shacham [16] obtained a construction in the random oracle model [10] with signatures shorter than 200 bytes [13].

In the BMW model, the population of users is frozen after the setup phase beyond which no new member can be added. Dynamic group signatures were independently formalized by Kiayias and Yung [45] and Bellare-Shi-Zhang [11]. In these models, pairing-based schemes with relatively short signatures were put forth in [54, 33]. Ateniese *et al.* [6] also gave a construction without random oracles using interactive assumptions. In the BMW model [9], Boyen and Waters independently came up with a different standard model proposal [19] using more classical assumptions and they subsequently refined their scheme [21] to

* This author was supported by the Belgian Fund for Scientific Research (F.R.S.-F.N.R.S.) via a "Chargé de recherches" fellowship.

** Supported by the IUAP B-Crypt Project and the Walloon Region Camus Project.

obtain constant-size signatures. In the dynamic model [11], Groth [38] described a system with constant-size signatures without random oracles but this scheme was rather a feasibility result than an efficient construction. Later on, Groth gave [39] a fairly efficient realization – with signatures consisting of about 50 group elements – in the standard model with the strongest anonymity level.

Revocation. In group signatures, membership revocation has received much attention in the last decade [22, 8, 29, 18] since revocation is central to digital signature schemes. One simple solution is to generate a new group public key and deliver a new signing key to each unrevoked member. However, in large groups, it may be inconvenient to change the public key and send a new secret to signers after they joined the group. An alternative approach taken by Bresson and Stern [22] is to have the signer prove that his membership certificate does not appear in a public list or revoked certificates. Unfortunately, the signer's workload and the size of signatures grow with the number of expelled users.

Song [55] presented an approach handling revocation in forward-secure group signatures. However, verification takes linear time in the number of excluded users.

Using accumulators[1] [12], Camenisch and Lysyanskaya [29] proposed a method (notably followed by [60, 27]) to revoke users in the ACJT group signature [7] while keeping $O(1)$ costs for signing and verifying. While elegant, this approach is history-dependent and requires users to keep track of all changes in the population of the group: at each modification of the accumulator value, unrevoked users need to update their membership certificates before signing new messages, which may require $O(r)$ exponentiations – if r is the number of revoked users – in the worst case.

Brickell [23] suggested the notion of *verifier-local revocation* group signatures, which was formalized by Boneh and Shacham [18] and further studied in [50, 61, 48]. In their systems, revocation messages are only sent to verifiers (making the signing algorithm independent of the number of revocations). The group manager maintains a revocation list (RL) which is used by verifiers to make sure that signatures were not generated by a revoked member. The RL contains a token for each revoked user and the verification algorithm has to verify signatures w.r.t. each token (a similar revocation mechanism is used in [24]). As a result, the verification cost is inevitably linear in the number of expelled users.

More recently, Nakanishi, Fuji, Hira and Funabiki [49] described a construction with constant complexities for signing/verifying and where group members never have to update their credentials. On the other hand, their proposal has the disadvantage of linear-size group public keys (in the maximal number N of users), although a tweak allows reducing the size to $O(N^{1/2})$.

In the context of anonymous credentials, Tsang *et al.* [58, 59] showed how to blacklist users without compromising their anonymity or involving a trusted third party. Their protocols either have linear proving complexity in the number of revocations or rely on accumulators (which may be problematic for our purposes). Camenisch, Kohlweiss and Soriente [28] suggested to handle revocations by periodically updating users credentials in which a specific attribute indicates a validity period. While useful in certain applications of anonymous credentials, in group signatures, their technique would place quite a burden on the group manager who would have to generate updates for each unrevoked individual credential.

1.2 Our Contribution

For the time being and despite over a decade of research efforts, group signatures in the standard model have no revocation mechanism allowing for scalable (*i.e.*, constant or polylogarithmic) verification time without dramatically degrading the efficiency in other metrics and without being history-dependent. In pairing-based group signatures, accumulator-based approaches are unlikely to result in solutions supporting very large groups. The reason is that, in known pairing-based accumulators [53, 27], public keys have linear size in the maximal number of accumulated values (unless one sacrifices the constant size of proofs of

[1] An accumulator allows hashing a set of values into a short string of constant size while allowing to efficiently prove that a specific value was accumulated.

non-membership as in [5]), which would result in linear-size group public keys in straightforward implementations. Recently [35], Fan *et al.* suggested a different way to use the accumulator of [27] and announced constant-size group public keys but their scheme still requires the group manager to publicize $O(N)$ values at each revocation. In a revocation mechanism along the lines of [29], Boneh, Boyen and Shacham [16] managed to avoid linear dependencies. However, their technique seems hard to combine[2] with Groth-Sahai proofs [40] so as to work in the standard model, which is our goal. In addition, we would like to save unrevoked users from having to update their keys after each revocation. To this end, it seems possible to adapt the approach of Nakanishi *et al.* [49] in the standard model. However, merely replacing sigma-protocols by Groth-Sahai proofs in the scheme of [49] would result in group public keys of size $O(N^{1/2})$ in the best case.

In this paper, we describe a novel and scalable revocation technique that interacts nicely with Groth-Sahai proofs and gives constructions in the standard model with $O(1)$ verification cost and at most polylogarithmic complexity in other metrics. Our approach bears similarities with the one of Nakanishi *et al.* [49] in that it does not require users to update their membership certificates at any time but, unlike [49], our group public key size is either $O(\log N)$ or constant. Like the scheme of [49], our main system uses revocation lists (RLs) of size $O(r)$ – which is in line with certificate revocation lists of standard PKIs – and we emphasize that these are *not* part of the group public key: verifiers only need to know the number of the latest revocation epoch and they do not have to read RLs entirely.

To obtain our constructions, we turn to the area of broadcast encryption and build on the Subset Cover framework of Naor, Naor and Lotspiech [51] (NNL). In a nutshell, the idea is to use the NNL ciphertext as a revocation list and have non-revoked signers prove their ability to decrypt in order to convince verifiers that they are not revoked. In its public-key variant, due to Dodis and Fazio [34], the Subset Cover framework relies on hierarchical identity-based encryption (HIBE) [44, 37] and each NNL ciphertext consists of several HIBE encryptions. To anonymously sign a message, we let group members commit to the specific HIBE ciphertext that they can decrypt (which gives constant-size signatures since only one ciphertext is committed to), and provide a non-interactive proof that: (i) they hold a private key which decrypts the committed HIBE ciphertext. (ii) The latter belongs to the revocation list.

By applying this approach to the Subset Difference (SD) method [51], we obtain a scheme with $O(1)$-size signatures, $O(\log N)$-size group public keys, membership certificates of size $O(\log^3 N)$ and revocation lists of size $O(r)$. The Layered Subset Difference method [41] can be used in the same way to obtain membership certificates of size $O(\log^{2.5} N)$ while retaining the same efficiency elsewhere. Using the Complete Subtree method, we also obtain a tradeoff with $O(r \cdot \log N)$ revocation lists, log-size membership certificates and constant-size group public keys (comparisons among schemes are given in Section 4).

A natural question is whether our SD-based revocable group signatures can generically use any HIBE scheme. The answer is negative as the Boneh-Boyen-Goh (BBG) construction [15] is currently the only suitable candidate. For anonymity reasons, ciphertexts should be of constant size and our security proof requires the HIBE system to satisfy a new and non-standard security property which is met by [15]. As we will see, the proof can hardly rely on the standard security notion for HIBE schemes [37].

We note that the new revocation mechanism can find applications in contexts other than group signatures. For example, it seems that it can be used in the oblivious transfer with access control protocol of [26], which also uses the technique of Nakanishi *et al.* [49] to revoke credentials.

[2] In the scheme of [16], signing keys consist of pairs $(g^{1/(\omega+s)}, s) \in \mathbb{G} \times \mathbb{Z}_p$, where $\omega \in \mathbb{Z}_p$ is the private key of the group manager, and the revocation mechanism relies on the availability of the exponent $s \in \mathbb{Z}_p$. In the standard model, the Groth-Sahai techniques would require to turn the membership certificates into triples $(g^{1/(\omega+s)}, g^s, u^s)$, for some $u \in \mathbb{G}$ (as in [21]), which is no longer compatible with the revocation technique.

2 Background

2.1 Bilinear Maps and Complexity Assumptions

We use bilinear maps $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ over groups of prime order p where $e(g,h) \neq 1_{\mathbb{G}_T}$ if and only if $g,h \neq 1_{\mathbb{G}}$. In these groups, we rely on hardness assumptions that are all falsifiable [52].

Definition 1 ([16]). *The* **Decision Linear Problem** *(DLIN) in* $\mathbb{G}$, *is to distinguish the distributions* $(g^a, g^b, g^{ac}, g^{bd}, g^{c+d})$ *and* $(g^a, g^b, g^{ac}, g^{bd}, g^z)$, *with* $a,b,c,d \overset{R}{\leftarrow} \mathbb{Z}_p^*$, $z \overset{R}{\leftarrow} \mathbb{Z}_p^*$. *The* **Decision Linear Assumption** *is the intractability of DLIN for any PPT distinguisher* D.

Definition 2 ([13]). *The* q**-Strong Diffie-Hellman problem** *(q-SDH) in* $\mathbb{G}$ *is, given* $(g, g^a, \ldots, g^{(a^q)})$, *for some* $g \overset{R}{\leftarrow} \mathbb{G}$ *and* $a \overset{R}{\leftarrow} \mathbb{Z}_p$, *to find a pair* $(g^{1/(a+s)}, s) \in \mathbb{G} \times \mathbb{Z}_p$.

Finally, we appeal to yet another "q-type" assumption introduced by Abe *et al.* [2].

Definition 3 ([2]). *In a group* $\mathbb{G}$, *the* q**-Simultaneous Flexible Pairing Problem** *(q-SFP) is, given* $\big(g_z,\ h_z,\ g_r,\ h_r,\ a,\ \tilde{a},\ b,\ \tilde{b} \in \mathbb{G}\big)$ *and* $q \in \mathsf{poly}(\lambda)$ *tuples* $(z_j, r_j, s_j, t_j, u_j, v_j, w_j) \in \mathbb{G}^7$ *such that*

$$e(a,\tilde{a}) = e(g_z, z_j) \cdot e(g_r, r_j) \cdot e(s_j, t_j) \quad \text{and} \quad e(b,\tilde{b}) = e(h_z, z_j) \cdot e(h_r, u_j) \cdot e(v_j, w_j), \tag{1}$$

to find a new tuple $(z^\star, r^\star, s^\star, t^\star, u^\star, v^\star, w^\star) \in \mathbb{G}^7$ *satisfying relation (1) and such that* $z^\star \neq 1_{\mathbb{G}}$ *and* $z^\star \neq z_j$ *for* $j \in \{1, \ldots, q\}$.

2.2 Groth-Sahai Proof Systems

In the following notations, for equal-dimension vectors or matrices A and B containing group elements, $A \odot B$ stands for their entry-wise product.

In their instantiations based on the DLIN assumption, the Groth-Sahai (GS) techniques [40] make use of prime order groups and a common reference string comprising vectors $\vec{f_1}, \vec{f_2}, \vec{f_3} \in \mathbb{G}^3$, where $\vec{f_1} = (f_1, 1, g)$, $\vec{f_2} = (1, f_2, g)$ for some $f_1, f_2 \in \mathbb{G}$. To commit to an element $X \in \mathbb{G}$, one sets $\vec{C} = (1,1,X) \odot \vec{f_1}^{\,r} \odot \vec{f_2}^{\,s} \odot \vec{f_3}^{\,t}$ with $r,s,t \overset{R}{\leftarrow} \mathbb{Z}_p^*$. When the CRS is configured to give perfectly sound proofs, we have $\vec{f_3} = \vec{f_1}^{\,\xi_1} \odot \vec{f_2}^{\,\xi_2}$ where $\xi_1, \xi_2 \in \mathbb{Z}_p^*$. Commitments $\vec{C} = (f_1^{r+\xi_1 t}, f_2^{s+\xi_2 t}, X \cdot g^{r+s+t(\xi_1+\xi_2)})$ are then Boneh-Boyen-Shacham (BBS) ciphertexts [16] that can be decrypted using $\beta_1 = \log_g(f_1)$, $\beta_2 = \log_g(f_2)$. In the witness indistinguishability (WI) setting, vectors $\vec{f_1}, \vec{f_2}, \vec{f_3}$ are linearly independent and $\vec{C}$ is a perfectly hiding commitment. Under the DLIN assumption, the two kinds of CRS are computationally indistinguishable.

To commit to a scalar $x \in \mathbb{Z}_p$, one computes $\vec{C} = \vec{\varphi}^{x} \odot \vec{f_1}^{\,r} \odot \vec{f_2}^{\,s}$, where $r,s \overset{R}{\leftarrow} \mathbb{Z}_p^*$, using a CRS comprising vectors $\vec{\varphi}, \vec{f_1}, \vec{f_2}$. In the soundness setting, $\vec{\varphi}, \vec{f_1}, \vec{f_2}$ are linearly independent (typically $\vec{\varphi} = \vec{f_3} \odot (1,1,g)$ where $\vec{f_3} = \vec{f_1}^{\,\xi_1} \odot \vec{f_2}^{\,\xi_2}$) whereas, in the WI setting, choosing $\vec{\varphi} = \vec{f_1}^{\,\xi_1} \odot \vec{f_2}^{\,\xi_2}$ gives a perfectly hiding commitment since $\vec{C}$ is always a BBS encryption of $1_{\mathbb{G}}$, no matter which exponent x is committed to.

To prove that committed variables satisfy a set of relations, the prover computes one commitment per variable and one proof element (made of a constant number of group elements) per relation.

Such proofs are available for pairing-product equations, which are relations of the type

$$\prod_{i=1}^{n} e(\mathcal{A}_i, \mathcal{X}_i) \cdot \prod_{i=1}^{n} \cdot \prod_{j=1}^{n} e(\mathcal{X}_i, \mathcal{X}_j)^{a_{ij}} = t_T, \tag{2}$$

for variables $\mathcal{X}_1, \ldots, \mathcal{X}_n \in \mathbb{G}$ and constants $t_T \in \mathbb{G}_T$, $\mathcal{A}_1, \ldots, \mathcal{A}_n \in \mathbb{G}$, $a_{ij} \in \mathbb{Z}_p$, for $i,j \in \{1, \ldots, n\}$. Efficient proofs also exist for multi-exponentiation equations

$$\prod_{i=1}^{m} \mathcal{A}_i^{y_i} \cdot \prod_{j=1}^{n} \mathcal{X}_j^{b_j} \cdot \prod_{i=1}^{m} \cdot \prod_{j=1}^{n} \mathcal{X}_j^{y_i \gamma_{ij}} = T, \tag{3}$$

for variables $\mathcal{X}_1,\ldots,\mathcal{X}_n \in \mathbb{G}$, $y_1,\ldots,y_m \in \mathbb{Z}_p$ and constants $T,\mathcal{A}_1,\ldots,\mathcal{A}_m \in \mathbb{G}$, $b_1,\ldots,b_n \in \mathbb{Z}_p$ and $\gamma_{ij} \in \mathbb{G}$, for $i \in \{1,\ldots,m\}, j \in \{1,\ldots,n\}$.

In pairing-product equations, proofs for quadratic equations require 9 group elements whereas linear equations (*i.e.*, where $a_{ij} = 0$ for all i,j in equation (2)) only take 3 group elements each. Linear multi-exponentiation equations of the type $\prod_{i=1}^{m} \mathcal{A}_i^{y_i} = T$ demand 2 group elements.

Multi-exponentiation equations admit zero-knowledge (NIZK) proofs at no additional cost. On a simulated CRS (prepared for the WI setting), a trapdoor makes it is possible to simulate proofs without knowing witnesses and simulated proofs have the same distribution as real proofs. In contrast, pairing-product equations do not always have NIZK proofs. Fortunately, NIWI proofs will be sufficient here.

2.3 Structure-Preserving Signatures

Several applications (see [2, 3, 36, 31, 4] for examples) require to sign groups elements while preserving the feasibility of efficiently proving that a committed signature is valid for a committed group element.

In [2, 3], Abe, Haralambiev and Ohkubo showed how to conveniently sign n group elements at once using signatures consisting of $O(1)$ group elements. Their scheme (which is referred to as the AHO signature in the paper) makes use of bilinear groups of prime order. In the context of symmetric pairings, the description below assumes public parameters $\mathsf{pp} = \big((\mathbb{G},\mathbb{G}_T),\ g\big)$ consisting of groups $(\mathbb{G},\mathbb{G}_T)$ of order $p > 2^\lambda$, where $\lambda \in \mathbb{N}$ is a security parameter, with a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ and a generator $g \in \mathbb{G}$.

Keygen(pp, n)**:** given an upper bound $n \in \mathbb{N}$ on the number of group elements that can be signed altogether, choose generators $G_r, H_r \stackrel{R}{\leftarrow} \mathbb{G}$. Pick $\gamma_z, \delta_z \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and $\gamma_i, \delta_i \stackrel{R}{\leftarrow} \mathbb{Z}_p$, for $i = 1$ to n. Then, compute $G_z = G_r^{\gamma_z}$, $H_z = H_r^{\delta_z}$ and $G_i = G_r^{\gamma_i}$, $H_i = H_r^{\delta_i}$ for each $i \in \{1,\ldots,n\}$. Finally, choose exponents $\alpha_a, \alpha_b \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and define $A = e(G_r, g^{\alpha_a})$ and $B = e(H_r, g^{\alpha_b})$. Set the public key as

$$pk = \big(G_r,\ H_r,\ G_z,\ H_z,\ \{G_i, H_i\}_{i=1}^n,\ A,\ B\big) \in \mathbb{G}^{2n+4} \times \mathbb{G}_T^2$$

while the private key consists of $sk = \big(\alpha_a, \alpha_b, \gamma_z, \delta_z, \{\gamma_i, \delta_i\}_{i=1}^n\big)$.

Sign$(sk, (M_1,\ldots,M_n))$**:** to sign a vector $(M_1,\ldots,M_n) \in \mathbb{G}^n$ using $sk = (\alpha_a, \alpha_b, \gamma_z, \delta_z, \{\gamma_i, \delta_i\}_{i=1}^n)$, choose $\zeta, \rho, \tau, \nu, \omega \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and compute $\theta_1 = g^\zeta$ as well as

$$\theta_2 = g^{\rho - \gamma_z \zeta} \cdot \prod_{i=1}^{n} M_i^{-\gamma_i}, \qquad \theta_3 = G_r^\tau, \qquad \theta_4 = g^{(\alpha_a - \rho)/\tau},$$
$$\theta_5 = g^{\nu - \delta_z \zeta} \cdot \prod_{i=1}^{n} M_i^{-\delta_i}, \qquad \theta_6 = H_r^\omega, \qquad \theta_7 = g^{(\alpha_b - \nu)/\omega},$$

The signature consists of $\sigma = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7)$.

Verify$(pk, \sigma, (M_1,\ldots,M_n))$**:** parse σ as $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7) \in \mathbb{G}^7$ and return 1 iff these equalities hold:

$$A = e(G_z, \theta_1) \cdot e(G_r, \theta_2) \cdot e(\theta_3, \theta_4) \cdot \prod_{i=1}^{n} e(G_i, M_i), \tag{4}$$
$$B = e(H_z, \theta_1) \cdot e(H_r, \theta_5) \cdot e(\theta_6, \theta_7) \cdot \prod_{i=1}^{n} e(H_i, M_i). \tag{5}$$

In [2, 3], the scheme was proved to be existentially unforgeable under chosen-message attacks under the q-SFP assumption, where q is the maximal number of signing queries.

Abe *et al.* [2, 3] also showed that signatures can be publicly randomized to obtain a different signature $\{\theta_i'\}_{i=1}^7 \leftarrow \mathsf{ReRand}(pk, \sigma)$ on $(M_1, \ldots, M_n)$. After randomization, we have $\theta_1' = \theta_1$ while $\{\theta_i'\}_{i=2}^7$ are uniformly distributed among the values satisfying the equalities $e(G_r, \theta_2') \cdot e(\theta_3', \theta_4') = e(G_r, \theta_2) \cdot e(\theta_3, \theta_4)$ and $e(H_r, \theta_5') \cdot e(\theta_6', \theta_7') = e(H_r, \theta_5) \cdot e(\theta_6, \theta_7)$. Moreover, $\{\theta_i'\}_{i\in\{3,4,6,7\}}$ are statistically independent of $(M_1, \ldots, M_n)$ and the rest of the signature. This implies that, in anonymity-related protocols, re-randomized $\{\theta_i'\}_{i\in\{3,4,6,7\}}$ can be safely revealed as long as $(M_1, \ldots, M_n)$ and $\{\theta_i'\}_{i\in\{1,2,5\}}$ are given in committed form.

In [4], Abe, Groth, Haralambiev and Ohkubo described a more efficient structure-preserving signature based on interactive assumptions. Here, we use the scheme of [2, 3] so as to rely on falsifiable assumptions.

2.4 The NNL Framework for Broadcast Encryption

The Subset Cover framework [51] considers secret-key broadcast encryption schemes with $N = 2^\ell$ registered receivers. Each one of them is associated with a leaf of a complete binary tree T of height ℓ and each tree node is assigned a secret key. If $\mathcal{N}$ denotes the universe of users and $\mathcal{R} \subset \mathcal{N}$ is the set of revoked receivers, the idea of the framework is to partition the set of non-revoked users into m disjoint subsets $S_1, \ldots, S_m$ such that $\mathcal{N} \backslash \mathcal{R} = S_1 \cup \ldots \cup S_m$. Depending on the way to partition $\mathcal{N} \backslash \mathcal{R}$ and the distribution of keys to users, different instantiations and tradeoffs are possible.

The Complete Subtree Method. In this technique, each subset S_i consists of the leaves of a complete subtree rooted at some node x_i of T. Upon registration, each user obtains secret keys for all nodes on the path connecting his leaf to the root of T (and thus $O(\ell)$ keys overall). By doing so, users in $\mathcal{N} \backslash \mathcal{R}$ can decrypt the content if the latter is enciphered using symmetric keys $K_1, \ldots, K_m$ corresponding to the roots of subtrees $S_1, \ldots, S_m$. As showed in [51], the CS partitioning method entails at most $m \leq r \cdot \log(N/r)$ subsets, where $r = |\mathcal{R}|$. Each transmission requires to send $O(r \cdot \log N)$ symmetric encryptions while, at each user, the storage complexity is $O(\log N)$.

As noted in [51, 34], a single-level identity-based encryption scheme allows implementing a public-key variant of the CS method. The master public key of the IBE scheme forms the public key of the broadcast encryption system, which allows for public keys of size $O(1)$ (instead of $O(N)$ in instantiations using ordinary public-key encryption). When users join the system, they obtain $O(\ell)$ IBE private keys (in place of symmetric keys) associated with the "identities" of nodes on the path between their leaf and the root.

The Subset Difference Method. The SD method reduces the transmission cost to $O(r)$ at the expense of increased storage requirements. For each node $x_j \in \mathsf{T}$, we call T_{x_j} the subtree rooted at x_j. The set $\mathcal{N} \backslash \mathcal{R}$ is now divided into disjoint subsets $S_{k_1,u_1}, \ldots, S_{k_m,u_m}$. For each $i \in \{1, \ldots, m\}$, the subset S_{k_i,u_i} is determined by a node x_{k_i} and one of its descendants x_{u_i} – which are called *primary* and *secondary* roots of S_{k_i,u_i}, respectively – and it consists of the leaves of $\mathsf{T}_{x_{k_i}}$ that are not in $\mathsf{T}_{x_{u_i}}$. Each user thus belongs to much more generic subsets than in the CS method and this allows reducing the maximal number of subsets to $m = 2r - 1$ (see [51] for a proof of this bound).

A more complex key distribution is necessary to avoid a prohibitive storage overhead. Each subset S_{k_i,u_i} is assigned a "proto-key" $P_{x_{k_i},x_{u_i}}$ that allows deriving the actual symmetric encryption key K_{k_i,u_i} for S_{k_i,u_i} and as well as proto-keys $P_{x_{k_i},x_{u_l}}$ for any descendant x_{u_l} of x_{u_i}. At the same time, $P_{x_{k_i},x_{u_l}}$ should be hard to compute without a proto-key $P_{x_{k_i},x_{u_i}}$ for an ancestor x_{u_i} of x_{u_l}. The key distribution phase then proceeds as follows. Let user i be assigned a leaf v_i and let $\epsilon = x_0, x_1, \ldots, x_\ell = v_i$ denote the path from the root ϵ to v_i. For each subtree T_{x_j} (with $j \in \{1, \ldots, \ell\}$), if copath_{x_j} denotes the set of all siblings of nodes on the path from x_j to v_i, user i must obtain proto-keys $P_{x_j,w}$ for each node $w \in \mathsf{copath}_{x_j}$ because he belongs to the generic subset whose primary root is x_j and whose secondary root is w. By storing $O(\ell^2)$ proto-keys (*i.e.*, $O(\ell)$ for each subtree T_{x_j}), users will be able to derive keys for all generic subsets they belong to.

In [34], Dodis and Fazio extended the SD method to the public-key setting using hierarchical identity-based encryption. In the tree, each node w at depth $\leq \ell$ has a label $\langle w \rangle$ which is defined by assigning the label ε to the root (at depth 0). The left and right children of w are then labeled with $\langle w \rangle || 0$ and $\langle w \rangle || 1$, respectively. For each subset S_{k_i,u_i} of $\mathcal{N} \backslash \mathcal{R}$, the sender considers the primary and secondary roots x_{k_i}, x_{u_i} and parses the label $\langle x_{u_i} \rangle$ as $\langle x_{k_i} \rangle || u_{i,\ell_{i,1}} \ldots u_{i,\ell_{i,2}}$, with $u_{i,j} \in \{0,1\}$ for each $j \in \{\ell_{i,1}, \ldots, \ell_{i,2}\}$. Then, he computes a HIBE ciphertext for the hierarchical identity $(\langle x_{k_i} \rangle, u_{i,\ell_{i,1}}, \ldots, u_{i,\ell_{i,2}})$ at level $\ell_{i,2} - \ell_{i,1} + 2$. Upon registration, if $\epsilon = x_0, \ldots, x_\ell = v_i$ denotes the path from the root to his leaf v_i, for each subtree T_{x_j}, user i receives exactly one HIBE private key for each $w \in \mathsf{copath}_{x_j}$: namely, for each $w \in \mathsf{copath}_{x_j}$, there exist $\ell_1, \ell_2 \in \{1, \ldots, \ell\}$ such that $\langle w \rangle = \langle x_j \rangle || w_{\ell_1} \ldots w_{\ell_2}$ with $w_j \in \{0,1\}$ for all $j \in \{\ell_1, \ldots, \ell_2\}$ and user i obtains a HIBE private key for the hierarchical identity $(\langle x_j \rangle, w_{\ell_1}, \ldots, w_{\ell_2})$. By construction, this key will allow user i to decrypt any HIBE ciphertext encrypted for a subset whose primary root is x_j and whose secondary root is a descendant of w. Overall, each user thus has to store $O(\log^2 N)$ HIBE private keys.

2.5 Revocable Group Signatures

We consider group signature schemes that have their lifetime divided into revocation epochs at the beginning of which group managers update their revocation lists.

The syntax and the security model are similar to [49] but they build on those defined by Kiayias and Yung [45]. Like the Bellare-Shi-Zhang model [11], the latter assumes an interactive join protocol between the group manager and the prospective user. This protocol provides the user with a membership certificate and a membership secret. Such protocols may consist of several rounds of interaction.

Syntax. We denote by $N \in \mathsf{poly}(\lambda)$ the maximal number of group members. At the beginning of each revocation epoch t, the group manager publicizes an up-to-date revocation list RL_t and we denote by $\mathcal{R}_t \subset \{1, \ldots, N\}$ the corresponding set of revoked users (we assume that $\mathcal{R}_t$ is part of RL_t). A revocable group signature (R-GS) scheme consists of the following algorithms or protocols.

Setup(λ, N)**:** given a security parameter $\lambda \in \mathbb{N}$ and a maximal number of group members $N \in \mathbb{N}$, this algorithm (which is run by a trusted party) generates a group public key $\mathcal{Y}$, the group manager's private key $\mathcal{S}_{\mathsf{GM}}$ and the opening authority's private key $\mathcal{S}_{\mathsf{OA}}$. Keys $\mathcal{S}_{\mathsf{GM}}$ and $\mathcal{S}_{\mathsf{OA}}$ are given to the appropriate authority while $\mathcal{Y}$ is publicized. The algorithm also initializes a public state St comprising a set data structure $St_{users} = \emptyset$ and a string data structure $St_{\mathsf{trans}} = \epsilon$.

Join: is an interactive protocol between the group manager GM and a user $\mathcal{U}_i$ where the latter becomes a group member. The protocol involves two interactive Turing machines $\mathsf{J}_{\mathsf{user}}$ and J_{GM} that both take as input $\mathcal{Y}$. The execution, denoted as $[\mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(\lambda, St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})]$, terminates with user $\mathcal{U}_i$ obtaining a membership secret sec_i, that no one else knows, and a membership certificate cert_i. If the protocol successfully terminates, the group manager updates the public state St by setting $St_{users} := St_{users} \cup \{i\}$ as well as $St_{\mathsf{trans}} := St_{\mathsf{trans}} || \langle i, \mathsf{transcript}_i \rangle$.

Revoke: is a (possibly randomized) algorithm allowing the GM to generate an updated revocation list RL_t for the new revocation epoch t. It takes as input a public key $\mathcal{Y}$ and a set $\mathcal{R}_t \subset St_{users}$ that identifies the users to be revoked. It outputs an updated revocation list RL_t for epoch t.

Sign: given a revocation epoch t with its revocation list RL_t, a membership certificate cert_i, a membership secret sec_i and a message M, this algorithm outputs $\perp$ if $i \in \mathcal{R}_t$ and a signature σ otherwise.

Verify: given a signature σ, a revocation epoch t, the corresponding revocation list RL_t, a message M and a group public key $\mathcal{Y}$, this deterministic algorithm returns either 0 or 1.

Open: takes as input a message M, a valid signature σ w.r.t. $\mathcal{Y}$ for the indicated revocation epoch t, the opening authority's private key $\mathcal{S}_{\mathsf{OA}}$ and the public state St. It outputs $i \in St_{users} \cup \{\perp\}$, which is the identity of a group member or a symbol indicating an opening failure.

Each membership certificate contains a unique tag that identifies the user.

A R-GS scheme must satisfy three security notions defined in appendix A. The first one is called *security against misidentification attacks.* It requires that, even if the adversary can introduce and revoke users at will, it cannot produce a signature that traces outside the set of unrevoked adversarially-controlled users.

As in ordinary (*i.e.*, non-revocable) group signatures, the notion of *security against framing attacks* mandates that, even if the whole system colludes against a user, that user will not bear responsibility for messages that he did not sign. Finally, the notion of *anonymity* is also defined (in the presence of a signature opening oracle) as in the models of [11, 45].

3 A Revocable Group Signature Based on the Subset Difference Method

As already mentioned, the idea is to turn the NNL global ciphertext into a revocation list in the group signature. Each group member is associated with a leaf of a binary tree of height ℓ and the outcome of the join protocol is the user obtaining a membership certificate that contains the same key material as in the public-key variant of the SD method (*i.e.*, $O(\ell^2)$ HIBE private keys). To ensure traceability and non-frameability, these NNL private keys are linked to a group element X, that only the user knows the discrete logarithm of, by means of structure-preserving signatures.

At each revocation epoch t, the group manager generates an up-to-date revocation list RL_t consisting of $O(r)$ HIBE ciphertexts, each of which is signed using a structure-preserving signature. When it comes to sign a message, the user $\mathcal{U}_i$ proves that he is not revoked by providing evidence that he is capable of decrypting one of the HIBE ciphertexts in RL_t. To this end, $\mathcal{U}_i$ commits to that HIBE ciphertext C_l and proves that he holds a key that decrypts C_l. To convince the verifier that C_l belongs to RL_t, he proves knowledge of a signature on the committed HIBE ciphertext C_l (this technique is borrowed from the set membership and range proofs of [57, 25]). Of course, to preserve the anonymity of signers, we need a HIBE scheme with constant-size ciphertexts (otherwise, the length of the committed ciphertext could betray the signer's location in the tree), which is why the Boneh-Boyen-Goh construction [15] is the ideal candidate.

The scheme is made anonymous and non-frameable using the same techniques as Groth [39] in steps 4-6 of the signing algorithm. As for the security against misidentification attacks, we cannot prove it by relying on the standard collusion-resistance (captured by Definition 7 in appendix B.1) of the HIBE scheme. In the proof of Theorem 1, the problem appears in the treatment of forgeries that open to a revoked user: while this user cannot have obtained a private key that decrypts the committed HIBE ciphertext of the forgery (because he is revoked), unrevoked adversarially-controlled users can. To solve this problem, we need to rest on a non-standard security property (formalized by Definition 8 in appendix B.1) called "key-robustness". This notion asks that, given a private key generated for some hierarchical identity using specific random coins, it be infeasible to compute the private key of a different identity for the *same random coins* and even *knowing* the *master secret key* of the HIBE scheme. While unusual, this property can be proved (by Lemma 1 in appendix B.2) under the standard Diffie-Hellman assumption for the BBG construction.

Perhaps surprisingly, even though we rely on the BBG HIBE, we do not need its underlying q-type assumption [15]. The reason is that the master secret key of the scheme is unnecessary here as its role is taken over by the private key of a structure-preserving signature. In the ordinary BBG system (recalled in appendix B.2), private keys contain components of the form $(g_2^\alpha \cdot F(\mathsf{ID})^r, g^r)$, for some $r \in \mathbb{Z}_p$, where g_2^α is the master secret key and $F(\mathsf{ID})$ is a function of the hierarchical identity. In the join protocol, the master key g_2^α disappears: the user obtains a private key of the form $(F(\mathsf{ID})^r, g^r)$ and an AHO signature is used to bind the user's membership public key X to g^r. The latter can be thought of as a public key for a one-time variant (the one-time nature is what allows for a proof of selective-message security in the standard model) of the Boneh-Lynn-Shacham signature [17]. The underlying one-time private key $r \in \mathbb{Z}_p$

is used to compute $F(\mathsf{ID})^r$ as well as a number of delegation components allowing to derive signatures for messages that ID is a prefix of (somewhat in the fashion of wildcard signatures [1][Section 6]).

3.1 Construction

As in Section 2.4, $\langle x \rangle$ denotes the label of node $x \in \mathsf{T}$ and, for any sub-tree T_{x_j} rooted at x_j and any leaf v_i of T_{x_j}, copath_{x_j} denotes the set of all siblings of nodes on the path from x_j to v_i, not counting x_j itself.

As is standard in group signatures, the description below assumes that, before joining the group, user $\mathcal{U}_i$ chooses a long term key pair $(\mathsf{usk}[i], \mathsf{upk}[i])$ and registers it in some PKI.

Setup(λ, N)**:** given a security parameter $\lambda \in \mathbb{N}$ and the permitted number of users $N = 2^\ell$,

1. Choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$, with a generator $g \stackrel{R}{\leftarrow} \mathbb{G}$.
2. Generate two key pairs $(sk^{(0)}_{\mathsf{AHO}}, pk^{(0)}_{\mathsf{AHO}})$ and $(sk^{(1)}_{\mathsf{AHO}}, pk^{(1)}_{\mathsf{AHO}})$ for the AHO signature in order to sign messages of two group elements. These key pairs consist of
$$pk^{(d)}_{\mathsf{AHO}} = \Big(G_r^{(d)},\ H_r^{(d)},\ G_z^{(d)} = G_r^{\gamma_z^{(d)}},\ H_z^{(d)} = H_r^{\delta_z^{(d)}},\ \{G_i^{(d)} = G_r^{\gamma_i^{(d)}}, H_i^{(d)} = H_r^{\delta_i^{(d)}}\}_{i=1}^2,\ A^{(d)},\ B^{(d)}\Big)$$
and $sk^{(d)}_{\mathsf{AHO}} = \big(\alpha_a^{(d)}, \alpha_b^{(d)}, \gamma_z^{(d)}, \delta_z^{(d)}, \{\gamma_i^{(d)}, \delta_i^{(d)}\}_{i=1}^2\big)$, where $d \in \{0,1\}$.
3. As a CRS for the NIWI proof system, select vectors $\mathbf{f} = (\vec{f_1}, \vec{f_2}, \vec{f_3})$ s.t. $\vec{f_1} = (f_1, 1, g) \in \mathbb{G}^3$, $\vec{f_2} = (1, f_2, g) \in \mathbb{G}^3$, and $\vec{f_3} = \vec{f_1}^{\,\xi_1} \cdot \vec{f_2}^{\,\xi_2}$, with $f_1 = g^{\beta_1}, f_2 = g^{\beta_2} \stackrel{R}{\leftarrow} \mathbb{G}$ and $\beta_1, \beta_2, \xi_1, \xi_2 \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$.
4. Choose $(U, V) \stackrel{R}{\leftarrow} \mathbb{G}^2$ that, together with f_1, f_2, g, will form a public encryption key.
5. Generate a master public key mpk_{BBG} for the Boneh-Boyen-Goh HIBE. Such a public key consists[3] of $mpk_{\mathsf{BBG}} = \big(\ \{h_i\}_{i=0}^{\ell}\big)$, where $\ell = \log_2(N)$, and no master secret key is needed.
6. Select an injective encoding[4] function $\mathcal{H} : \{0,1\}^{\leq \ell} \rightarrow \mathbb{Z}_p^*$ and a strongly unforgeable one-time signature $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$.
7. Set $\mathcal{S}_{\mathsf{GM}} := \big(sk^{(0)}_{\mathsf{AHO}}, sk^{(1)}_{\mathsf{AHO}}\big)$, $\mathcal{S}_{\mathsf{OA}} := \big(\beta_1, \beta_2\big)$ as authorities' private keys and the group public key is
$$\mathcal{Y} := \Big(g,\ pk^{(0)}_{\mathsf{AHO}},\ pk^{(1)}_{\mathsf{AHO}},\ mpk_{\mathsf{BBG}},\ \mathbf{f},\ (U,V),\ \mathcal{H},\ \Sigma\Big).$$

Join$^{(\mathrm{GM},\mathcal{U}_i)}$**:** the GM and the prospective user $\mathcal{U}_i$ run the following protocol $[\mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(\lambda, St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})]$:

1. $\mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y})$ picks $x \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and computes $X = g^x$ which is sent to $\mathsf{J}_{\mathsf{GM}}(\lambda, St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})$. If the value X already appears in some entry $\mathsf{transcript}_j$ of the database St_{trans}, J_{GM} aborts and returns $\perp$ to $\mathsf{J}_{\mathsf{user}}$.
2. J_{GM} assigns to $\mathcal{U}_i$ an available leaf v_i of label $\langle v_i \rangle = v_{i,1} \ldots v_{i,\ell} \in \{0,1\}^\ell$ in the tree T. Let $x_0 = \epsilon$, $x_1, \ldots, x_{\ell-1}, x_\ell = v_i$ be the path from v_i to the root ϵ of T. For $j = 0$ to ℓ, J_{GM} does the following.
   a. Consider the sub-tree T_{x_j} rooted at node x_j. Let copath_{x_j} be the co-path from x_j to v_i.
   b. For each node $w \in \mathsf{copath}_{x_j}$, since x_j is an ancestor of w, $\langle x_j \rangle$ is a prefix of $\langle w \rangle$ and we denote by $w_{\ell_1} \ldots w_{\ell_2} \in \{0,1\}^{\ell_2 - \ell_1 + 1}$, for some $\ell_1 \leq \ell_2 \leq \ell$, the suffix of $\langle w \rangle$ coming right after $\langle x_j \rangle$.

[3] As mentioned earlier, in comparison with the original HIBE scheme (recalled in appendix B.2) where mpk_{BBG} includes $(g_1 = g^\alpha, g_2)$ and $msk_{\mathsf{BBG}} = g_2^\alpha$, the public elements g_1 and g_2 have disappeared.

[4] This encoding allows making sure that "identities" will be non-zero at each level. Since the set $\{0,1\}^{\leq \ell}$ is of cardinality $\sum_{i=0}^{\ell} 2^i = 2^{\ell+1} - 1 < p - 1$, such a function can be efficiently constructed without any intractability assumption.

b.1 Choose a random $r \xleftarrow{R} \mathbb{Z}_p$ and compute a HIBE private key

$$\begin{aligned} d_w &= (D_{w,1}, D_{w,2}, K_{w,\ell_2-\ell_1+3}, \ldots, K_{w,\ell}) \\ &= \Big(\big(h_0 \cdot h_1^{\mathcal{H}(\langle x_j \rangle)} \cdot h_2^{\mathcal{H}(w_{\ell_1})} \cdots h_{\ell_2-\ell_1+2}^{\mathcal{H}(w_{\ell_2})}\big)^r,\ g^r,\ h_{\ell_2-\ell_1+3}^r, \ldots,\ h_\ell^r \Big) \end{aligned}$$

for the identity $(\mathcal{H}(\langle x_j \rangle), \mathcal{H}(w_{\ell_1}), \ldots, \mathcal{H}(w_{\ell_2})) \in (\mathbb{Z}_p^*)^{\ell_2-\ell_1+2}$.

b.2 Using $sk_{\mathsf{AHO}}^{(0)}$, generate an AHO signature $\sigma_w = (\theta_{w,1}, \ldots, \theta_{w,7})$ on $(X, D_{w,2}) \in \mathbb{G}^2$ so as to bind the HIBE private key d_w to the value X that identifies $\mathcal{U}_i$.

3. J_{GM} sends $\langle v_i \rangle \in \{0,1\}^\ell$, and the HIBE private keys $\{\{d_w\}_{w \in \mathsf{copath}_{x_j}}\}_{j=0}^{\ell}$ to $\mathsf{J}_{\mathsf{user}}$ that verifies their validity. If these keys are all well-formed, $\mathsf{J}_{\mathsf{user}}$ acknowledges them by generating a digital signature $sig_i = \mathrm{Sign}_{\mathsf{usk}[i]}\big(X || \{\{d_w\}_{w \in \mathsf{copath}_{x_j}}\}_{j=0}^{\ell}\big)$ and sends it back to J_{GM}.
4. J_{GM} checks that $\mathrm{Verify}_{\mathsf{upk}[i]}\big(X || \{\{d_w\}_{w \in \mathsf{copath}_{x_j}}\}_{j=0}^{\ell}, sig_i\big) = 1$. If not J_{GM} aborts. Otherwise, J_{GM} returns the AHO signatures $\{\{\sigma_w\}_{w \in \mathsf{copath}_{x_j}}\}_{j=0}^{\ell}$ to $\mathsf{J}_{\mathsf{user}}$ and stores the conversation transcript $\mathsf{transcript}_i = (X, \{\{d_w, \sigma_w\}_{w \in \mathsf{copath}_{x_j}}\}_{j=0}^{\ell}, sig_i)$ in the database St_{trans}.
5. $\mathsf{J}_{\mathsf{user}}$ defines the membership certificate cert_i as $\mathsf{cert}_i = \big(\langle v_i \rangle, \{\{d_w, \sigma_w\}_{w \in \mathsf{copath}_{x_j}}\}_{j=0}^{\ell}, X\big)$, where X will serve as the tag that identifies $\mathcal{U}_i$. The membership secret sec_i is defined to be $\mathsf{sec}_i = x$.

Revoke$(\mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, t, \mathcal{R}_t)$**:**

1. Parse $\mathcal{S}_{\mathsf{GM}}$ as $\mathcal{S}_{\mathsf{GM}} := \big(sk_{\mathsf{AHO}}^{(0)}, sk_{\mathsf{AHO}}^{(1)}\big)$.
2. Using the SD covering algorithm, find a cover of the unrevoked user set $\{1, \ldots, N\} \backslash \mathcal{R}_t$ as the union of disjoint subsets of the form $S_{k_1,u_1}, \ldots, S_{k_m,u_m}$, with $m \leq 2 \cdot |\mathcal{R}_t| - 1$.
3. For $i = 1$ to m, do the following.

   a. Consider S_{k_i,u_i} as the difference between sub-trees rooted at an internal node x_{k_i} and one of its descendants x_{u_i}. The label of x_{u_i} can be written $\langle x_{u_i} \rangle = \langle x_{k_i} \rangle || u_{i,\ell_{i,1}} \ldots u_{i,\ell_{i,2}}$ for some $\ell_{i,1} < \ell_{i,2} \leq \ell$ and where $u_{i,\kappa} \in \{0,1\}$ for each $\kappa \in \{\ell_{i,1}, \ldots, \ell_{i,2}\}$. Then, compute an encoding of S_{k_i,u_i} as a group element

   $$C_i = h_0 \cdot h_1^{\mathcal{H}(\langle x_{k_i} \rangle)} \cdot h_2^{\mathcal{H}(u_{i,\ell_{i,1}})} \cdots h_{\ell_{i,2}-\ell_{i,1}+2}^{\mathcal{H}(u_{i,\ell_{i,2}})}.$$

   Note that C_i can be thought of as a de-randomized HIBE ciphertext for the hierarchical identity $\big(\mathcal{H}(\langle x_{k_i} \rangle), \mathcal{H}(u_{i,\ell_{i,1}}), \ldots, \mathcal{H}(u_{i,\ell_{i,2}})\big) \in (\mathbb{Z}_p^*)^{\ell_{i,2}-\ell_{i,1}+2}$.

   b. To authenticate the HIBE ciphertext C_i and bind it to the revocation epoch t, use $sk_{\mathsf{AHO}}^{(1)}$ to generate an AHO signature $\Theta_i = (\Theta_{i,1}, \ldots, \Theta_{i,7}) \in \mathbb{G}^7$ on the pair $(C_i, g^t) \in \mathbb{G}^2$, where the epoch number t is interpreted as an element of $\mathbb{Z}_p$.

Return the revocation data RL_t which is defined to be

$$RL_t = \Big(t,\ \mathcal{R}_t,\ \{\langle x_{k_i} \rangle,\ \langle x_{u_i} \rangle,\ \big(C_i, \Theta_i = (\Theta_{i,1}, \ldots, \Theta_{i,7})\big)\}_{i=1}^m\Big) \tag{6}$$

Sign$(\mathcal{Y}, t, RL_t, \mathsf{cert}_i, \mathsf{sec}_i, M)$**:** return $\perp$ if $i \in \mathcal{R}_t$. Otherwise, to sign $M \in \{0,1\}^*$, generate a one-time signature key pair $(\mathsf{SK}, \mathsf{VK}) \leftarrow \mathcal{G}(\lambda)$. Parse cert_i as $\big(\langle v_i \rangle, \{\{(d_w, \sigma_w)\}_{w \in \mathsf{copath}_{x_j}}\}_{j=0}^{\ell}, X\big)$ and sec_i as $x \in \mathbb{Z}_p$. Then, $\mathcal{U}_i$ conducts the following steps.

1. Using RL_t, determine the set S_{k_l,u_l}, with $l \in \{1, \ldots, m\}$, that contains the leaf v_i (this subset must exist since $i \notin \mathcal{R}_t$) and let x_{k_l} and x_{u_l} denote the primary and secondary roots of S_{k_l,u_l}. Since

x_{k_l} is an ancestor of x_{u_l}, we can write $\langle x_{u_l} \rangle = \langle x_{k_l} \rangle || u_{l,\ell_1} \dots u_{l,\ell_2}$, for some $\ell_1 < \ell_2 \leq \ell$ and with $u_{l,\kappa} \in \{0,1\}$ for each $\kappa \in \{\ell_1, \dots, \ell_2\}$. The signer $\mathcal{U}_i$ computes a HIBE decryption key of the form

$$(D_{l,1}, D_{l,2}) = \Big(\big(h_0 \cdot h_1^{\mathcal{H}(\langle x_{k_l} \rangle)} \cdot h_2^{\mathcal{H}(u_{l,\ell_1})} \cdots h_{\ell_2-\ell_1+2}^{\mathcal{H}(u_{l,\ell_2})}\big)^r,\ g^r \Big). \tag{7}$$

This is possible since, if we denote by $\langle x_{k,l} \rangle || u_{l,\ell_1} \dots u_{l,\ell_1'}$ the shortest prefix of $\langle x_{u_l} \rangle$ that is not a prefix of $\langle v_i \rangle$, the key material $\{d_w\}_{w \in \mathsf{copath}_{x_{k_l}}}$ corresponding to the sub-tree rooted at x_{k_l} contains a HIBE private key $d_w = (D_{w,1}, D_{w,2}, K_{w,\ell_1'-\ell_1+3}, \dots, K_{w,\ell})$ such that

$$d_w = \Big(\big(h_0 \cdot h_1^{\mathcal{H}(\langle x_{k_l} \rangle)} \cdot h_2^{\mathcal{H}(u_{l,\ell_1})} \cdots h_{\ell_1'-\ell_1+2}^{\mathcal{H}(u_{l,\ell_1'})}\big)^r,\ g^r,\ h_{\ell_1'-\ell_1+3}^r, \dots, h_\ell^r \Big),$$

which allows deriving a key of the form (7) for the same $r \in \mathbb{Z}_p$ (*i.e.*, $D_{l,2} = D_{w,2}$).

2. To prove his ability to "decrypt" C_l, $\mathcal{U}_i$ first re-randomizes Θ_l as $\{\Theta_{l,i}'\}_{i=1}^7 \leftarrow \mathsf{ReRand}(pk_{\mathsf{AHO}}^{(1)}, \Theta_l)$. Then, he computes a Groth-Sahai commitment com_{C_l} to C_l as well as commitments $\{com_{\Theta_{l,i}'}\}_{i \in \{1,2,5\}}$ to $\{\Theta_{l,i}'\}_{i \in \{1,2,5\}}$. He generates a proof π_{C_l} that C_l is a certified HIBE ciphertext for epoch t: *i.e.*, π_{C_l} provides evidence that

$$\begin{aligned} A^{(1)} \cdot e(\Theta_{l,3}', \Theta_{l,4}')^{-1} \cdot e(G_2^{(1)}, g^t)^{-1} &= e(G_z^{(1)}, \Theta_{l,1}') \cdot e(G_r^{(1)}, \Theta_{l,2}') \cdot e(G_1^{(1)}, C_l), \\ B^{(1)} \cdot e(\Theta_{l,6}', \Theta_{l,7}')^{-1} \cdot e(H_2^{(1)}, g^t)^{-1} &= e(H_z^{(1)}, \Theta_{l,1}') \cdot e(H_r^{(1)}, \Theta_{l,5}') \cdot e(H_1^{(1)}, C_l), \end{aligned} \tag{8}$$

Then, $\mathcal{U}_i$ generates commitments $\{com_{D_{l,i}}\}_{i=1}^2$ to the HIBE key components $\{D_{l,i}\}_{i=1}^2$ derived at step 1 and computes a proof π_{D_l} that $e(D_{l,1}, g) = e(C_l, D_{l,2})$. The latter is quadratic and requires 9 group elements. Since $\{\Theta_{l,i}'\}_{i \in \{3,4,6,7\}}$ are constants, equations (8) are linear and require 3 elements each. Hence, π_{C_l} and π_{D_l} take 15 elements altogether.

3. Let $\sigma_l = (\theta_{l,1}, \dots, \theta_{l,7})$ be the AHO signature on $(X, D_{l,2})$. Compute $\{\theta_{l,i}'\}_{i=1}^7 \leftarrow \mathsf{ReRand}(pk_{\mathsf{AHO}}^{(0)}, \sigma_l)$ and generate commitments $\{com_{\theta_{l,i}'}\}_{i \in \{1,2,5\}}$ to $\{\theta_{l,i}'\}_{i \in \{1,2,5\}}$ as well as a commitment com_X to X. Then, generate a proof π_{σ_l} that committed variables satisfy the verification equations

$$\begin{aligned} A^{(0)} \cdot e(\theta_{l,3}', \theta_{l,4}')^{-1} &= e(G_z^{(0)}, \theta_{l,1}') \cdot e(G_r^{(0)}, \theta_{l,2}') \cdot e(G_1^{(0)}, X) \cdot e(G_2^{(0)}, D_{l,2}), \\ B^{(0)} \cdot e(\theta_{l,6}', \theta_{l,7}')^{-1} &= e(H_z^{(0)}, \theta_{l,1}) \cdot e(H_r^{(0)}, \theta_{l,5}') \cdot e(H_1^{(0)}, X) \cdot e(H_2^{(0)}, D_{l,2}) \end{aligned}$$

Since these equations are linear, π_{σ_l} requires 6 group elements.

4. Using VK as a tag (we assume that it is first hashed onto $\mathbb{Z}_p$ in such a way that it can be interpreted as a $\mathbb{Z}_p$ element), compute a tag-based encryption [47] of X by drawing $z_1, z_2 \xleftarrow{R} \mathbb{Z}_p$ and setting

$$(\Psi_1, \Psi_2, \Psi_3, \Psi_4, \Psi_5) = \big(f_1^{z_1}, f_2^{z_2}, X \cdot g^{z_1+z_2}, (g^{\mathsf{VK}} \cdot U)^{z_1}, (g^{\mathsf{VK}} \cdot V)^{z_2}\big).$$

5. Generate a NIZK proof that $com_X = (1,1,X) \cdot \vec{f_1}^{\phi_{X,1}} \cdot \vec{f_2}^{\phi_{X,2}} \cdot \vec{f_3}^{\phi_{X,3}}$ and (Ψ_1, Ψ_2, Ψ_3) are BBS encryptions of the same value X. If we write $\vec{f_3} = (f_{3,1}, f_{3,2}, f_{3,3})$, the Groth-Sahai commitment com_X can be written as $(f_1^{\phi_{X,1}} \cdot f_{3,1}^{\phi_{X,3}}, f_2^{\phi_{X,2}} \cdot f_{3,2}^{\phi_{X,3}}, X \cdot g^{\phi_{X,1}+\phi_{X,2}} \cdot f_{3,3}^{\phi_{X,3}})$, so that we have

$$com_X \odot (\Psi_1, \Psi_2, \Psi_3)^{-1} = \big(f_1^{\tau_1} \cdot f_{3,1}^{\tau_3},\ f_2^{\tau_2} \cdot f_{3,2}^{\tau_3},\ g^{\tau_1+\tau_2} \cdot f_{3,3}^{\tau_3}\big) \tag{9}$$

with $\tau_1 = \phi_{X,1} - z_1$, $\tau_2 = \phi_{X,2} - z_2$, $\tau_3 = \phi_{X,3}$. The signer $\mathcal{U}_i$ commits to $\tau_1, \tau_2, \tau_3 \in \mathbb{Z}_p$ (by computing $com_{\tau_j} = \vec{\varphi}^{\tau_j} \cdot \vec{f_1}^{\phi_{\tau_j,1}} \cdot \vec{f_2}^{\phi_{\tau_j,2}}$, for $j \in \{1,2,3\}$, using the vector $\vec{\varphi} = \vec{f_3} \cdot (1,1,g)$ and random $\{\phi_{\tau_j,1}, \phi_{\tau_j,2}\}_{j=1}^3$), and generates proofs $\{\pi_{eq\text{-}com,j}\}_{j=1}^3$ that τ_1, τ_2, τ_3 satisfy the three relations (9). Since these are linear equations, proofs $\{\pi_{eq\text{-}com,j}\}_{j=1}^3$ cost 2 elements each.

6. Compute $\sigma_{\mathsf{VK}} = g^{1/(x+\mathsf{VK})}$ and generate a commitment $com_{\sigma_{\mathsf{VK}}}$ to σ_{VK}. Then, generate a NIWI proof that committed variables σ_{VK} and X satisfy $e(\sigma_{\mathsf{VK}}, X \cdot g^{\mathsf{VK}}) = e(g,g)$. This relation is quadratic and requires a proof consisting of 9 group elements. We denote this proof by $\pi_{\sigma_{\mathsf{VK}}} = (\vec{\pi}_{\sigma_{\mathsf{VK}},1}, \vec{\pi}_{\sigma_{\mathsf{VK}},2}, \vec{\pi}_{\sigma_{\mathsf{VK}},3})$.
7. Compute $\sigma_{ots} = \mathcal{S}(\mathsf{SK}, (M, RL_t, \Psi_1, \Psi_2, \Psi_3, \Psi_4, \Psi_5, \Omega, \mathbf{com}, \mathbf{\Pi}))$ where $\Omega = \{\Theta'_{l,i}, \theta'_{l,i}\}_{i\in\{3,4,6,7\}}$ and

$$\mathbf{com} = \big(com_{C_l}, \{com_{D_{l,i}}\}_{i=1}^{2}, com_X, \{com_{\Theta'_{l,i}}\}_{i\in\{1,2,5\}}, \{com_{\theta'_{l,i}}\}_{i\in\{1,2,5\}}, \{com_{\tau_i}\}_{i=1}^{3}, com_{\sigma_{\mathsf{VK}}}\big)$$
$$\mathbf{\Pi} = (\pi_{C_l}, \pi_{D_l}, \pi_{\sigma_l}, \pi_{eq\text{-}com,1}, \pi_{eq\text{-}com,2}, \pi_{eq\text{-}com,3}, \pi_{\sigma_{\mathsf{VK}}})$$

Return the signature $\sigma = \big(\mathsf{VK}, \Psi_1, \Psi_2, \Psi_3, \Psi_4, \Psi_5, \Omega, \mathbf{com}, \mathbf{\Pi}, \sigma_{ots}\big)$.

Verify$(\sigma, M, t, RL_t, \mathcal{Y})$**:** parse σ as above and do the following.

1. If $\mathcal{V}(\mathsf{VK}, (M, RL_t, \Psi_1, \Psi_2, \Psi_3, \Psi_4, \Psi_5, \Omega, \mathbf{com}, \mathbf{\Pi}), \sigma_{ots}) = 0$, return 0.
2. Return 0 if $e(\Psi_1, g^{\mathsf{VK}} \cdot U) \neq e(f_1, \Psi_4)$ or $e(\Psi_2, g^{\mathsf{VK}} \cdot V) \neq e(f_2, \Psi_5)$.
3. Return 1 if all proofs properly verify. Otherwise, return 0.

Open$(M, t, RL_t, \sigma, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, St)$**:** given $\mathcal{S}_{\mathsf{OA}} = (\beta_1, \beta_2)$, parse the signature σ as above and return $\perp$ if $\mathsf{Verify}(\sigma, M, t, RL_t, \mathcal{Y}) = 0$. Otherwise, compute $\tilde{X} = \Psi_3 \cdot \Psi_1^{-1/\beta_1} \cdot \Psi_2^{-1/\beta_2}$. In the database St_{trans}, find a record $\langle i, \mathsf{transcript}_i = (X, \{\{d_w, \sigma_w\}_{w\in\mathsf{copath}_{x_j}}\}_{j=0}^{\ell}, sig_i)\rangle$ such that $X = \tilde{X}$. If no such record exists in St_{trans}, return $\perp$. Otherwise, return i.

From an efficiency point of view, for each $i \in \{1\ldots, m\}$, RL_t comprises 8 group elements plus the labels of nodes that identify S_{k_i,u_i}. If $\lambda_{\mathbb{G}}$ denotes the bitlength of a group element, the number of bits of RL_t is thus bounded by $2 \cdot |\mathcal{R}_t| \cdot (8 \cdot \lambda_{\mathbb{G}} + 2\log N) < 2 \cdot |\mathcal{R}_t| \cdot (9\lambda_{\mathbb{G}})$ bits (as $\log N < \lambda_{\mathbb{G}}/2$ since $\lambda \leq \lambda_{\mathbb{G}}$ and N is polynomial). The size of revocation lists thus amounts to that of at most $18 \cdot |\mathcal{R}_t|$ group elements.

Group members need $O(\log^3 N)$ group elements to store their membership certificate. As far as the size of signatures goes, **com** and $\mathbf{\Pi}$ require 42 and 36 group elements, respectively. If the one-time signature of [38] is used, σ consists of 96 group elements, which is less than twice the size of Groth's signatures [39]. At the 128-bit security level, if each element has a representation of 512 bits, a signature takes 6 kB.

Verifying signatures takes constant time. The cost of each signature generation is dominated by at most $\ell = \log N$ exponentiations to derive a HIBE private key at step 1. However, this step only has to be executed once per revocation epoch, at the first signature of that epoch.

3.2 Security

Theorem 1 (Misidentification). *The scheme is secure against misidentification attacks assuming that the q-SFP problem is hard for $q = \max(\ell^2 \cdot q_a, q_r^2)$, where q_a and q_r denote the maximal numbers of $Q_{\mathsf{a\text{-}join}}$ queries and Q_{revoke} queries, respectively, and $\ell = \log N$.*

Proof. To mount a successful misidentification attack, the adversary $\mathcal{A}$ must output a non-trivial signature for which the opening algorithm fails to point to an unrevoked adversarially-controlled group member.

Let $\sigma^\star = \big(\mathsf{VK}^\star, \Psi_1^\star, \Psi_2^\star, \Psi_3^\star, \Psi_4^\star, \Psi_5^\star, \Omega^\star, \mathbf{com}^\star, \mathbf{\Pi}^\star, \sigma_{ots}^\star\big)$ denote $\mathcal{A}$'s forgery and parse $\mathbf{com}^\star$ as

$$\mathbf{com}^\star = \Big(com_C^\star, \{com_{D_i}^\star\}_{i=1}^{2}, com_X^\star, \{com_{\Theta'_i}^\star\}_{i\in\{1,2,5\}}, \{com_{\theta'_i}^\star\}_{i\in\{1,2,5\}}, \{com_{\tau_i}^\star\}_{i=1}^{3}, com_{\sigma_{\mathsf{VK}^\star}}^\star\Big).$$

By hypothesis, it must be the case that $\mathsf{Open}(M^\star, t^\star, RL_{t^\star}, \sigma^\star, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, St) \notin U^a \backslash \mathcal{R}_{t^\star}$, where U^a denotes the set of adversarially-controlled users. Depending on extractable commitments $com_X^\star$, $com_C^\star$, $\{com_{D_i}^\star\}_{i=1}^{2}$, $\{com_{\Theta'_i}^\star\}_{i\in\{1,2,5\}}$, $\{com_{\theta'_i}^\star\}_{i\in\{1,2,5\}}$ and their contents, we distinguish the following cases:

- **Type I forgeries** are those for which $com^{\star}_{C_l}$ contains a group element $C^{\star}$ such that $(C^{\star}, g^{t^{\star}})$ was never signed when the latest revocation list $RL_{t^{\star}}$ was generated.
- **Type II forgeries** are such that $com^{\star}_{C}$ contains a properly certified HIBE ciphertext for epoch $t^{\star}$ (say $C^{\star} = C^{\star}_l$, for some $l \in \{1, \dots, m\}$, where $C^{\star}_1, \dots, C^{\star}_m$ are the HIBE ciphertexts of $RL_{t^{\star}}$). However, the execution of Open reveals a previously unseen $X^{\star}$ or points some revoked user $i \in U^a \cap \mathcal{R}_{t^{\star}}$ although $\sigma^{\star}$ provides convincing evidence that the committed private key $(D^{\star}_1, D^{\star}_2)$ allows decrypting $C^{\star}_l$ and that committed elements $\{\theta^{\star}_i\}_{i=1}^{7}$ form a valid signature on $(X^{\star}, D^{\star}_2)$. In this case, we have two situations:
  a. The pair $(X^{\star}, D^{\star}_2)$ was *not* signed by $\mathsf{J_{GM}}$ in any execution of Join. It means that either: (1) Open uncovers a value $X^{\star}$ that does not appear anywhere in St_{trans}. (2) The traced user $i \in U^a \cap \mathcal{R}_{t^{\star}}$ colluded with some unrevoked user $j \in U^a$ whose leaf *is* in S_{k_l,u_l} – which $C^{\star}_l$ is an encoding of – and managed to forge an AHO signature so as to link his $X^{\star}$ to an authorized key $(D^{\star}_1, D^{\star}_2)$ for S_{k_l,u_l}.
  b. The pair $(X^{\star}, D^{\star}_2)$ was signed by $\mathsf{J_{GM}}$ at some execution of Join. At first, we would like to use the Type II.b adversary to break the standard selective semantic security of the HIBE system (cf. Definition 7 in appendix B). As it turns out, even if we were using the original BBG HIBE (with its master secret key), such a reduction would be unlikely to work because the set S_{k_l,u_l} may contain unrevoked users in U^a, so that $\mathcal{A}$ obtained private keys that do decrypt $C^{\star}_l$. Instead, we rely on the security property that we call "key-robustness" (defined in appendix B.1) and which relies on a weaker assumption than the standard security of the BBG HIBE. Observe that, when user i joined the group, he cannot have been issued the private key $(D^{\star}_1, D^{\star}_2)$ (which is an authorized key for S_{k_l,u_l}) since he is revoked at epoch $t^{\star}$. However, since $(X^{\star}, D^{\star}_2)$ was signed by $\mathsf{J_{GM}}$, user i must have obtained from $\mathsf{J_{GM}}$ a HIBE private key of the form $(D_1, D^{\star}_2)$, where $D_1 \neq D^{\star}_1$, for an identity other than the one encoded by S_{k_l,u_l}. In this case, as will be showed in Lemma 2, the key-robustness property is necessarily broken in the HIBE scheme.

It is easy to see that Type I and Type II.a forgeries imply a forger against the AHO signature scheme (the proof is straightforward and omitted).

Lemma 2 (in appendix C) demonstrates that a Type II.b attack necessarily contradicts the key-robustness property (formally defined in appendix B.1) of the Boneh-Boyen-Goh HIBE scheme and thus the Diffie-Hellman assumption, as established by Lemma 1 in appendix B.2.

Finally, one can readily check that an adversary cannot produce a signature $\sigma^{\star}$ allowing to win the misidentification game without being one of the above kinds of forgeries. The result of the theorem follows from the fact that the SFP assumption implies the CDH assumption. □

The security against framing attacks and the anonymity property rely on the SDH and DLIN assumptions, respectively, and the proofs are given in appendices D.1 and D.2.

4 Efficiency Comparisons

This section discusses the comparative efficiency of known pairing-based revocable group signatures. We focus on revocation methods that are more efficient than generic revocation techniques: for example, we do not consider techniques (such as the one recalled in [19][Section 5.4]) consisting in privately sending new keys to all remaining users at each revocation. Also, we only consider schemes where group members are stateless and do not have to update their membership certificate every time a revocation occurs.

In table 1, all sizes are given in terms of number of group elements, each one of which costs $O(\lambda)$ bits to represent.

Table 1. Comparison between pairing-based revocable group signatures

Schemes	Group public key size	Signature size	Membership certificate size	Revocation list size	Signature cost	Verification cost	Revocation cost*	Standard model?
NFHF1 [49]	$O(N)$	$O(1)$	$O(1)$	$O(r)$	$O(1)$	$O(1)$	$O(r)$	✗
NFHF2 [49]	$O(N^{1/2})$	$O(1)$	$O(1)$	$O(r)$	$O(1)$	$O(1)$	$O(r)$	✗
BS [18]	$O(1)$	$O(1)$	$O(1)$	$O(r)$	$O(1)$	$O(r)$	$O(1)$	✗
NF [50]	$O(T)^{\diamondsuit}$	$O(1)$	$O(1)$	$O(r)$	$O(1)$	$O(r)$	$O(r)$	✗
LV [48]	$O(T)^{\diamondsuit}$	$O(1)$	$O(1)$	$O(r)$	$O(1)$	$O(r)$	$O(r)$	✓
This work (SD)	$O(\log N)$	$O(1)$	$O(\log^3 N)$	$O(r)$	$O(\log N)^{\dagger\ \ddagger}$	$O(1)$	$O(r \cdot \log N)^{\ddagger}$	✓
This work (CS)	$O(1)$	$O(1)$	$O(\log N)$	$O(r \cdot \log(N/r))$	$O(1)^{\ddagger}$	$O(1)$	$O(r \cdot \log(N/r))^{\ddagger}$	✓

N: max. number of users; r: number of revocations T: max. number of revocation epochs

$*$ The revocation cost refers to the complexity of generating an up-to-date revocation list.

$\diamondsuit$ These schemes can be modified to have $O(1)$-size group public keys.

† This complexity is only involved at the first signature of each revocation epoch.

‡ We only count arithmetic operations. In the signing algorithm, for example, we neglect $O(\log \log N)$ combinatorial operations at the beginning of each epoch.

As we can see, our CS and SD-based constructions are not only the first revocable group signatures with constant verification time in the standard model. Among schemes where revocations do not entail updates in unrevoked users' credentials, they are also the only solutions offering $O(1)$ verification cost and at most poly-logarithmic complexity in other metrics.

In applications where one can afford a logarithmic expansion factor in the size of revocation lists, the CS method seems preferable as it features compact (meaning logarithmic according to the terminology used in [9, 19]) membership certificates.

In situations where the size of the revocation list is to be minimized, the SD method should be preferred. Alternatively, the Layered Subset Difference approach (LSD) [41] provides an interesting tradeoff: at the expense of doubling the maximal size of revocation lists (which asymptotically remain of size $O(r)$), its basic variant allows reducing the size of membership certificates to $O(\log^{5/2} N)$ as only $O(\log^{3/2} N)$ HIBE private keys have to be stored.

Acknowledgements

We thank Damien Vergnaud, Sébastien Canard, Roch Lescuyer and the anonymous reviewers for useful comments.

References

1. M. Abdalla, E. Kiltz, G. Neven. Generalized Key Delegation for Hierarchical Identity-Based Encryption. In *ESORICS'07*, *LNCS* 4734, pp. 139–154. Springer, 2007.
2. M. Abe, K. Haralambiev, M. Ohkubo. Signing on Elements in Bilinear Groups for Modular Protocol Design. Cryptology ePrint Archive: Report 2010/133, 2010.
3. M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, M. Ohkubo. Structure-Preserving Signatures and Commitments to Group Elements. In *Crypto'10*, *LNCS* 6223, pp. 209–236, 2010.
4. M. Abe, J. Groth, K. Haralambiev, M. Ohkubo. Optimal Structure-Preserving Signatures in Asymmetric Bilinear Groups. In *Crypto'11*, *LNCS* 6841, pp. 649–666, 2011.
5. T. Acar, L. Nguyen. Revocation for Delegatable Anonymous Credentials. In *PKC'11*, *LNCS* 6571, pp. 423–440, 2011.
6. G. Ateniese, J. Camenisch, S. Hohenberger, B. de Medeiros. Practical group signatures without random oracles. Cryptology ePrint Archive: Report 2005/385, 2005.
7. G. Ateniese, J. Camenisch, M. Joye, G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *Crypto'00*, *LNCS* 1880, pp. 255–270, 2000.
8. G. Ateniese, D. Song, G. Tsudik. Quasi-Efficient Revocation in Group Signatures. In *Financial Cryptography'02*, *LNCS* 2357, pp. 183–197, 2002.

9. M. Bellare, D. Micciancio, B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *Eurocrypt'03*, *LNCS* 2656, pp. 614–629, 2003.
10. M. Bellare, P. Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *1st ACM Conference on Computer and Communications Security*, pp. 62–73, ACM Press, 1993.
11. M. Bellare, H. Shi, C. Zhang. Foundations of group signatures: The case of dynamic groups. In *CT-RSA'05*, *LNCS* 3376, pp. 136–153, 2005.
12. J. Benaloh, M. de Mare. One-Way Accumulators: A Decentralized Alternative to Digital Sinatures. In *Eurocrypt'93*, *LNCS* 4948, pp. 274–285, 1993.
13. D. Boneh, X. Boyen. Short Signatures Without Random Oracles. In *Eurocrypt'04*, *LNCS* 3027, pp. 56–73. Springer-Verlag, 2004.
14. D. Boneh, X. Boyen. Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In *Eurocrypt'04*, *LNCS* 3027, pp. 223–238, 2004.
15. D. Boneh, X. Boyen, E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In *Eurocrypt'05*, *LNCS* 3494, pp. 440–456, 2005.
16. D. Boneh, X. Boyen, H. Shacham. Short Group Signatures. In *Crypto'04*, *LNCS* 3152, pp. 41–55. Springer, 2004.
17. D. Boneh, B. Lynn, H. Shacham. Short signatures from the Weil pairing. In *Asiacrypt'01*, *LNCS* 2248, pp. 514–532. Springer, 2001.
18. D. Boneh, H. Shacham. Group signatures with verifier-local revocation. In *ACM-CCS'04*, pp. 168–177. ACM Press, 2004.
19. X. Boyen, B. Waters. Compact Group Signatures Without Random Oracles. In *Eurocrypt'06*, *LNCS* 4004, pp. 427–444, Springer, 2006.
20. X. Boyen, B. Waters. Anonymous Hierarchical Identity-Based Encryption (Without Random Oracles). In *Crypto'06*, *LNCS* 4117, pp. 290–307, 2006.
21. X. Boyen, B. Waters. Full-Domain Subgroup Hiding and Constant-Size Group Signatures. In *PKC'07*, *LNCS* 4450, pp. 1–15, 2007.
22. E. Bresson, J. Stern. Efficient Revocation in Group Signatures. In *PKC'01*, *LNCS* 1992, pp. 190–206, 2001.
23. E. Brickell. An efficient protocol for anonymously providing assurance of the container of the private key. Submission to the Trusted Computing Group. April, 2003.
24. E. Brickell, J. Camenisch, L. Chen. Direct Anonymous Attestation. In *ACM-CCS'04*, pp. 132–145, 2004.
25. J. Camenisch, R. Chaabouni, a. shelat. Efficient Protocols for Set Membership and Range Proofs. In *Asiacrypt'08*, *LNCS* 5350, pp. 234–252, Springer, 2008.
26. J. Camenisch, M. Dubovitskaya, G. Neven, G. Zaverucha. Oblivious Transfer with Hidden Access Control Policies. In *PKC'11*, *LNCS* 6571, pp. 192–209, 2011.
27. J. Camenisch, M. Kohlweiss, C. Soriente. An Accumulator Based on Bilinear Maps and Efficient Revocation for Anonymous Credentials. In *PKC'09*, *LNCS* 5443, pp. 481–500, 2009.
28. J. Camenisch, M. Kohlweiss, C. Soriente. Solving Revocation with Efficient Update of Anonymous Credentials. In *SCN'10*, *LNCS* 6280, pp. 454–471, 2010.
29. J. Camenisch, A. Lysyanskaya. Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. In *Crypto'02*, *LNCS* 2442, pp. 61–76, Springer, 2002.
30. R. Canetti, S. Halevi, J. Katz. A Forward-Secure Public-Key Encryption Scheme. In *Eurocrypt'03*, *LNCS* 2656, pp. 254–271, 2003.
31. J. Cathalo, B. Libert, M. Yung. Group Encryption: Non-Interactive Realization in the Standard Model. In *Asiacrypt'09*, *LNCS* 5912, pp. 179–196, 2009.
32. D. Chaum, E. van Heyst. Group Signatures. In *Eurocrypt'91*, *LNCS* 547, pp. 257–265, Springer, 1991.
33. C. Delerablée, D. Pointcheval. Dynamic Fully Anonymous Short Group Signatures. In *Vietcrypt'06*, *LNCS* 4341, pp. 193–210, Springer, 2006.
34. Y. Dodis, N. Fazio. Public Key Broadcast Encryption for Stateless Receivers. In *Digital Rights Management (DRM'02)*, *LNCS* 2696, pp. 61–80, 2002.
35. C.-I. Fan, R.-H. Hsu, M. Manulis. Group Signature with Constant Revocation Costs for Signers and Verifiers. In *Cryptology and Network Security (CANS 2011)*, *LNCS* 7092, pp. 214–233, 2011.
36. G. Fuchsbauer. Automorphic Signatures in Bilinear Groups and an Application to Round-Optimal Blind Signatures. Cryptology ePrint Archive: Report 2009/320, 2009.
37. C. Gentry, A. Silverberg. Hierarchical ID-based cryptography. In *Asiacrypt'02*, *LNCS* 2501, Springer, 2002.
38. J. Groth. Simulation-Sound NIZK Proofs for a Practical Language and Constant Size Group Signatures. In *Asiacrypt'06*, *LNCS* 4284, pp. 444–459, Springer, 2006.
39. J. Groth. Fully anonymous group signatures without random oracles. In *Asiacrypt 2007*, *LNCS* 4833, pp. 164–180. Springer, 2007.
40. J. Groth, A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Eurocrypt'08*, LNCS 4965, pp. 415–432, 2008.
41. D. Halevy, A. Shamir. The LSD broadcast encryption scheme. In *Crypto'02*, *LNCS* 2442, pp. 47–60, Springer, 2002.

42. D. Hofheinz, T. Jager, E. Kiltz. Short Signatures From Weaker Assumptions. In *Asiacrypt'11*, *LNCS* series, to appear, 2011.
43. D. Hofheinz, E. Kiltz. Programmable hash functions and their applications. In *Crypto'08*, *LNCS* 5157, pp. 21–38, 2008.
44. J. Horwitz, B. Lynn. Toward hierarchical identity-based encryption. In *Eurocrypt'02*, *LNCS* 2332, Springer, 2002.
45. A. Kiayias, M. Yung. Secure scalable group signature with dynamic joins and separable authorities. International Journal of Security and Networks (IJSN) Vol. 1, No. 1/2, pp. 24–45, 2006. Earlier version appeared as Cryptology ePrint Archive: Report 2004/076, 2004.
46. A. Kiayias, M. Yung. Group signatures with efficient concurrent join. In *Eurocrypt'05*, *LNCS* 3494, pp. 198–214, 2005.
47. E. Kiltz. Chosen-ciphertext security from tag-based encryption. In *TCC'06*, *LNCS* 3876, pp. 581–600, 2006.
48. B. Libert, D. Vergnaud. Group Signatures with Verifier-Local Revocation and Backward Unlinkability in the Standard Model. In *CANS'09*, *LNCS* 5888, pp. 498-517, 2009.
49. T. Nakanishi, H. Fujii, Y. Hira, N. Funabiki. Revocable Group Signature Schemes with Constant Costs for Signing and Verifying. In *PKC'09*, *LNCS* 5443, pp. 463–480, 2009.
50. T. Nakanishi, N. Funabiki. Verifier-Local Revocation Group Signature Schemes with Backward Unlinkability from Bilinear Maps. In *Asiacrypt'05*, *LNCS* 5443, pp. 533–548, 2009.
51. M. Naor, D. Naor, J. Lotspiech. Revocation and Tracing Schemes for Stateless Receivers. In *Crypto'01*, *LNCS* 2139, pp. 41–62, 2001.
52. M. Naor. On Cryptographic Assumptions and Challenges. In *Crypto'03*, *LNCS* 2729, pp. 96–109. Springer-Verlag, 2003.
53. L. Nguyen. Accumulators from Bilinear Pairings and Applications. In *CT-RSA'05*, *LNCS* 3376, pp. 275–292, 2005.
54. L. Nguyen, R. Safavi-Naini. Efficient and Provably Secure Trapdoor-Free Group Signature Schemes from Bilinear Pairings. In *Asiacrypt'04*, *LNCS* 3329, pp. 372–386. Springer-Verlag, 2004.
55. D. Song. Practical forward secure group signature schemes. In *ACM-CCS'01*, pp. 225–234, 2001.
56. V. Shoup. Lower bounds for discrete logarithms and related problems. In *Eurocrypt'97*, *LNCS* 1233, pp. 256–66, 1997.
57. I. Teranishi, K. Sako. k-Times Anonymous Authentication with a Constant Proving Cost. In *PKC'06*, *LNCS* 3958, pp. 525–542, Springer, 2006.
58. P. Tsang, M.-Ho Au, A. Kapadia, S. Smith. Blacklistable anonymous credentials: blocking misbehaving users without TTPs. In *ACM-CCS'07*, pp. 72–81, 2007.
59. P. Tsang, M.-Ho Au, A. Kapadia, S. Smith. PEREA: towards practical TTP-free revocation in anonymous authentication. In *ACM-CCS'08*, pp. 333–344, 2008.
60. G. Tsudik, S. Xu. Accumulating Composites and Improved Group Signing. In *Asiacrypt'03*, *LNCS* 2894, pp. 269–286, 2003.
61. S. Zhou, D. Lin. Shorter Verifier-Local Revocation Group Signatures from Bilinear Maps. In *CANS'06*, *LNCS* 4301, pp. 126–143, Springer, 2006.

A Correctness and Security Definitions for Revocable Group Signatures

In the following, a public state St is said *valid* if it can be reached from $St = (\emptyset, \varepsilon)$ by a Turing machine having oracle access to J_{GM}. Likewise, a state St' is said to *extend* another state St if it can be reached from St.

Similarly to [45, 46], we will write $\mathsf{cert}_i \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_i$ to express that there exist coin tosses ϖ for J_{GM} and J_{user} such that, for some valid public state St', the execution of $[\mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(\lambda, St', \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})](\varpi)$ provides $\mathsf{J}_{\mathsf{user}}$ with $\langle i, \mathsf{sec}_i, \mathsf{cert}_i \rangle$.

Correctness. We say that a R-GS scheme is correct if:

1. In a valid state St, it holds that $|St_{users}| = |St_{trans}|$ and no two entries of St_{trans} can contain certificates with the same tag.
2. If $[\mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(\lambda, St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})]$ is honestly run by both parties and $\langle i, \mathsf{cert}_i, \mathsf{sec}_i \rangle$ is obtained by $\mathsf{J}_{\mathsf{user}}$, then it holds that $\mathsf{cert}_i \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_i$.
3. For each revocation epoch t and any $\langle i, \mathsf{cert}_i, \mathsf{sec}_i \rangle$ such that $\mathsf{cert}_i \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_i$, satisfying condition 2, if $i \not\in \mathcal{R}_t$, it holds that $\mathsf{Verify}\big(\mathsf{Sign}(\mathcal{Y}, t, RL_t, \mathsf{cert}_i, \mathsf{sec}_i, M), M, t, RL_t, \mathcal{Y}\big) = 1$.
4. For any $\langle i, \mathsf{cert}_i, \mathsf{sec}_i \rangle$ resulting from the interaction $[\mathsf{J}_{\mathsf{user}}(.,.), \mathsf{J}_{\mathsf{GM}}(., St, ., .)]$ for some valid state St, any revocation epoch t such that $i \not\in \mathcal{R}_t$, if $\sigma = \mathsf{Sign}(\mathcal{Y}, t, RL_t, \mathsf{cert}_i, \mathsf{sec}_i, M)$, then

$$\mathsf{Open}(M, t, RL_t, \sigma, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, St') = i.$$

Security Model. As in [45], we formalize security properties via experiments where the adversary interacts with a stateful interface $\mathcal{I}$ that maintains the following variables:

- $\mathsf{state}_{\mathcal{I}}$: is a data structure representing the state of the interface as the adversary invokes oracles. It is initialized as $\mathsf{state}_{\mathcal{I}} = (St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(\lambda, N)$. It comprises the (initially empty) set St_{users} of group members and a database St_{trans} containing transcripts of join protocols. Finally, $\mathsf{state}_{\mathcal{I}}$ includes a counter t (which is initialized to 0) indicating the number of user revocation queries so far.
- $n = |St_{users}| < N$ is the current cardinality of the group.
- Sigs: is a database of signatures issued by the signing oracle. Each record is a triple (i, t, M, σ) indicating that message M was signed by user i during period t.
- U^a: is the set of users that are adversarially-controlled since their introduction in the system.
- U^b: is the set of honest users that were introduced by the adversary acting as a dishonest group manager. For such users, the adversary obtains the transcript of the join protocol but not the user's membership secret.

When mounting attacks, adversaries will be granted access to the following oracles.

- Q_{pub}, Q_{keyGM} and Q_{keyOA}: when these oracles are invoked, the interface looks up $\mathsf{state}_{\mathcal{I}}$ and returns the group public key $\mathcal{Y}$, the GM's private key $\mathcal{S}_{\mathsf{GM}}$ and the opening authority's private key $\mathcal{S}_{\mathsf{OA}}$ respectively.
- $Q_{\mathsf{a\text{-}join}}$: allows the adversary to introduce users under his control in the group. On behalf of the GM, the interface interacts with the malicious prospective user by running J_{GM} in the join protocol. If the protocol successfully terminates, the interface increments N, updates St by inserting the new user n in sets St_{users} and U^a. It also sets $St_{\mathsf{trans}} := St_{\mathsf{trans}} || \langle n, \mathsf{transcript}_n \rangle$.
- $Q_{\mathsf{b\text{-}join}}$: allows the adversary, acting as a dishonest group manager, to introduce new group members of his choice. The interface starts an execution of $[\mathsf{J}_{\mathsf{user}}, \mathsf{J}_{\mathsf{GM}}]$ and runs $\mathsf{J}_{\mathsf{user}}$ in interaction with the J_{GM}-executing adversary. If the protocol successfully completes, the interface increments n, adds user n to St_{users} and U^b and sets $St_{\mathsf{trans}} := St_{\mathsf{trans}} || \langle n, \mathsf{transcript}_n \rangle$. It stores the membership certificate cert_n and the membership secret sec_n in a *private* part of $\mathsf{state}_{\mathcal{I}}$.
- Q_{sig}: given a message M, an index i, the interface checks if the private area of $\mathsf{state}_{\mathcal{I}}$ contains a certificate cert_i and a membership secret sec_i such that $i \notin \mathcal{R}_t$, where t is the current revocation epoch. If no such elements exist or if $i \notin U^b$, it returns $\perp$. Otherwise, it generates a signature σ on behalf of user i for epoch t. It also sets $\mathsf{Sigs} \leftarrow \mathsf{Sigs} || (i, t, M, \sigma)$.
- Q_{open}: on input of a valid pair (M, σ) for some revocation epoch t, the interface runs the opening algorithm using the current state St. When S is a set of triples (M, σ, t), $Q_{\mathsf{open}}^{\neg S}$ denotes the restricted oracle that applies the opening procedure to any triple (M, σ, t) but those in S.
- Q_{read} and Q_{write}: allow the adversary to read and write the content of $\mathsf{state}_{\mathcal{I}}$. When invoked, Q_{read} outputs the whole $\mathsf{state}_{\mathcal{I}}$ but the public/private keys and the private part of $\mathsf{state}_{\mathcal{I}}$ where membership secrets are stored after $Q_{\mathsf{b\text{-}join}}$-queries. Queries Q_{write} allow the adversary to modify $\mathsf{state}_{\mathcal{I}}$ as long as it does not remove or alter elements of St_{users}, St_{trans} or invalidate the public state St: for example, the adversary can use it to create dummy users at will as long as it does not re-use already existing certificate tags.
- Q_{revoke}: is a user revocation oracle. On input of an index i such that $i \in St_{users}$, the interface checks if i is in the appropriate user set (*i.e.*, U^a or U^b depending on the considered security notion) and if St_{trans} contains a record $\langle i, \mathsf{transcript}_i \rangle$ such that $i \notin \mathcal{R}_t$, where t is the current revocation epoch. If not, it returns $\perp$. Otherwise, it increments t, adds i to $\mathcal{R}_t$ and generates an updated revocation list RL_t which is made available to the adversary. For simplicity, we assumed that the adversary only revokes one user per query to Q_{revoke} but the model easily extends to allow multiple revocations at once.

The KY model considers properties termed security against *misidentification attacks*, *framing attacks* and *anonymity*.

In a misidentification attack, the adversary is allowed to corrupt the opening authority via the Q_{keyOA} oracle. He can also introduce corrupt users in the group via $Q_{\mathsf{a\text{-}join}}$-queries and revoke users at will using Q_{revoke}. His goal is to produce a signature $\sigma^\star$ that verifies w.r.t. $RL_{t^\star}$, where $t^\star$ denotes the current revocation epoch (*i.e.*, the number of Q_{revoke}-queries). It wins if the produced signature $\sigma^\star$ does not open to any unrevoked adversarially-controlled.

Definition 4. *A R-GS scheme is secure against misidentification attacks if, for any PPT adversary $\mathcal{A}$ involved in the experiment hereafter, we have* $\mathbf{Adv}_{\mathcal{A}}^{\text{mis-id}}(\lambda) = \Pr[\mathbf{Expt}_{\mathcal{A}}^{\text{mis-id}}(\lambda) = 1] \in \mathsf{negl}(\lambda)$.

```
Experiment $\mathbf{Expt}_{\mathcal{A}}^{\text{mis-id}}(\lambda)$
    $\mathsf{state}_{\mathcal{I}} = (St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(\lambda, N)$;
    $(M^\star, \sigma^\star) \leftarrow \mathcal{A}(Q_{\mathsf{pub}}, Q_{\mathsf{a\text{-}join}}, Q_{\mathsf{revoke}}, Q_{\mathsf{read}}, Q_{\mathsf{keyOA}})$;
    If $\mathsf{Verify}(\sigma^\star, M, t^\star, RL_{t^\star}, \mathcal{Y}) = 0$ return $0$;
    $i = \mathsf{Open}(M^\star, t^\star, RL_{t^\star}, \sigma^\star, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, St')$;
    If $(i \notin U^a \backslash \mathcal{R}_{t^\star})$ return $1$;
    Return $0$;
```

This definition extends the usual definition [45] in that $\mathcal{A}$ is also successful if $\sigma^\star$ verifies w.r.t. $RL_{t^\star}$ but opens to an adversarially-controlled user that was revoked during the revocation epoch $t^\star$.

Framing attacks consider the situation where the whole system, including the group manager and the opening authority, conspires against some honest user. The adversary is allowed to corrupt the group manager *and* the opening authority (using Q_{keyGM} and Q_{keyOA}, respectively). He can also introduce honest group members (via $Q_{\mathsf{b\text{-}join}}$-queries), observe the system while these users generate signatures and create dummy users using Q_{write}. In addition, before the possible corruption of the group manager, the adversary can revoke group members at any time by invoking the Q_{revoke} oracle. As a potentially corrupted group manager, $\mathcal{A}$ is allowed to come up with his own revocation list $RL_{t^\star}$ at the end of the game. We assume that anyone can publicly verify that $RL_{t^\star}$ is correctly formed (*i.e.*, that it could be a legitimate output of Revoke) so that the adversary does not come up with an ill-formed revocation list. For consistency, if $\mathcal{A}$ chooses not to corrupt the GM, the produced revocation list $RL_{t^\star}$ must be the one determined by the history of Q_{revoke}-queries. The adversary eventually aims at framing an uncorrupt group member.

Definition 5. *A R-GS scheme is secure against framing attacks if, for any PPT adversary $\mathcal{A}$, it holds that* $\mathbf{Adv}_{\mathcal{A}}^{\text{fra}}(\lambda) = \Pr[\mathbf{Expt}_{\mathcal{A}}^{\text{fra}}(\lambda) = 1] \in \mathsf{negl}(\lambda)$.

```
Experiment $\mathbf{Expt}_{\mathcal{A}}^{\text{fra}}(\lambda)$
    $\mathsf{state}_{\mathcal{I}} = (St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(\lambda, N)$;
    $(M^\star, \sigma^\star, t^\star, RL_{t^\star}) \leftarrow \mathcal{A}(Q_{\mathsf{pub}}, Q_{\mathsf{keyGM}}, Q_{\mathsf{keyOA}}, Q_{\mathsf{b\text{-}join}}, Q_{\mathsf{revoke}}, Q_{\mathsf{sig}}, Q_{\mathsf{read}}, Q_{\mathsf{write}})$;
    If $\mathsf{Verify}(\sigma^\star, M^\star, t^\star, RL_{t^\star}, \mathcal{Y}) = 0$ then return $0$;
    $i = \mathsf{Open}(M^\star, t^\star, RL_{t^\star}, \sigma^\star, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, St')$;
    If $i \notin U^b$ return $0$;
    If $\big(\bigwedge_{j \in U^b \text{ s.t. } j=i} (j, t^\star, M^\star, *) \notin \mathsf{Sigs}\big)$ then return $1$;
    Return $0$;
```

Anonymity is defined via a game involving a 2-stage adversary. To give a proper definition, we need to define an special algorithm IsRevoked which, given a valid membership certificate/secret pair $(\mathsf{cert}, \mathsf{sec})$ and a revocation list RL_t, allows efficiently deciding if $(\mathsf{cert}, \mathsf{sec})$ belongs to a revoked user for RL_t. Such an algorithm exists in our construction.

In the first stage of the game, called play stage, the adversary is allowed to modify $\mathsf{state}_{\mathcal{I}}$ by making Q_{write}-queries and to open signatures of his choice by invoking Q_{open}. At the end of the play stage, it chooses a message-period pair $(M^\star, t^\star)$ and two pairs $(\mathsf{sec}_0^\star, \mathsf{cert}_0^\star)$, $(\mathsf{sec}_1^\star, \mathsf{cert}_1^\star)$, consisting of a well-formed membership certificate and a membership secret for $b = 0, 1$. Note that, to prevent the adversary from trivially winning, we impose the constraint $\mathsf{IsRevoked}(\mathsf{sec}_b^\star, \mathsf{cert}_b^\star, RL_{t^\star}) = 0$ for each $b \in \{0,1\}$. The challenger flips a fair binary coin $d \overset{R}{\leftarrow} \{0,1\}$ and generates a signature $\sigma^\star$ using $(\mathsf{sec}_d^\star, \mathsf{cert}_d^\star)$. The adversary aims to eventually determine the bit d. Of course, it is restricted not to query the opening of $(M^\star, \sigma^\star)$ during the guess stage.

Definition 6. *A R-GS scheme is fully anonymous if* $\mathbf{Adv}^{\mathrm{anon}}(\mathcal{A}) := |\Pr[\mathbf{Expt}_{\mathcal{A}}^{\mathrm{anon}}(\lambda) = 1] - 1/2|$ *is negligible for any PPT adversary* $\mathcal{A}$ *involved in the following experiment:*

```
Experiment $\mathbf{Expt}_{\mathcal{A}}^{\mathrm{anon}}(\lambda)$
    $\mathsf{state}_{\mathcal{I}} = (St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(\lambda)$;
    $\big(aux, M^\star, t^\star, RL_{t^\star}, (\mathsf{sec}_0^\star, \mathsf{cert}_0^\star), (\mathsf{sec}_1^\star, \mathsf{cert}_1^\star)\big)$
        $\leftarrow \mathcal{A}(\mathsf{play} : Q_{\mathsf{pub}}, Q_{\mathsf{keyGM}}, Q_{\mathsf{revoke}}, Q_{\mathsf{open}}, Q_{\mathsf{read}}, Q_{\mathsf{write}})$;
    If $\neg(\mathsf{cert}_b^\star \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_b^\star)$ or $\mathsf{IsRevoked}(\mathsf{sec}_b^\star, \mathsf{cert}_b^\star, RL_{t^\star}) = 1$ for $b \in \{0,1\}$
        or if $\mathsf{cert}_0^\star = \mathsf{cert}_1^\star$ return 0;
    $d \overset{R}{\leftarrow} \{0,1\}$;  $\sigma^\star \leftarrow \mathsf{Sign}(\mathcal{Y}, t^\star, \mathsf{cert}_d^\star, \mathsf{sec}_d^\star, M^\star)$;
    $d' \leftarrow \mathcal{A}(\mathsf{guess} : \sigma^\star, aux, Q_{\mathsf{pub}}, Q_{\mathsf{keyGM}}, Q_{\mathsf{open}}^{\neg\{(M^\star, \sigma^\star, t^\star)\}}, Q_{\mathsf{read}}, Q_{\mathsf{write}})$;
    If $d' = d$ then return 1;
    Return 0;
```

B Hierarchical Identity-Based Encryption

Consider a hierarchy of entities, each of which has a unique address $\mathsf{ID} = (I_1, \ldots, I_\ell)$, with $I_i \in \{0,1\}^*$ for $1 \leq i \leq \ell$, at level ℓ. For any $i \leq \ell$, $\mathsf{ID}|_i$ denotes the prefix $(I_1, \ldots, I_i)$ of ID. The address of a node at level i is obtained by appending its local identifier I_i to its father's address $\mathsf{ID}|_{i-1}$.

A HIBE scheme [44, 37] is a tuple $(\mathsf{Setup}, \mathsf{Keygen}, \mathsf{Derive}, \mathsf{Encrypt}, \mathsf{Decrypt})$ of algorithms[5] working as follows. Setup is run by a trusted private key generator (PKG) to generate a master public key mpk and a master secret key msk. The latter is used by the PKG, at the root of the hierarchy, to derive private keys from users' identities at level 1. The key generation algorithm Keygen takes as input the master secret key msk and a hierarchical identity $\mathsf{ID} = (I_1, \ldots, I_k)$ and returns a private key d_{ID} for that identity. Algorithm Derive is used by a ℓ-th level entity with address $\mathsf{ID} = (I_1, \ldots, I_\ell)$ to compute private keys for its children labeled as $(I_1, \ldots, I_\ell, *)$ at depth $\ell + 1$. It takes in a ℓ-th level private key d_{ID} and a vector $\mathsf{ID}' = (I_1, \ldots, I_\ell, I_{\ell+1})$, where $\mathsf{ID}'|_\ell = \mathsf{ID}$, to generate a $(\ell + 1)$-th level secret key $d_{\mathsf{ID}'}$. Algorithm $\mathsf{Encrypt}$ takes in a plaintext $m \in \mathcal{M}$, where $\mathcal{M}$ denotes the plaintext space, the master public key mpk and the receiver's address $\mathsf{ID} = (I_1, \ldots, I_\ell)$ to produce a ciphertext C that can be undone by the receiver having obtained d_{ID} from its father.

In many HIBE constructions (such as [15, 20]) delegated private keys (produced by Derive) have the same distribution as original keys (generated by Keygen). In the upcoming security definitions, we assume that this property is satisfied by the considered HIBE system.

In the following, we say that a HIBE scheme is *key-partitioned* if private keys $d_{\mathsf{ID}} = (D_{\mathsf{ID}}, K_{\mathsf{ID}})$ consist of two distinct part: the first one D_{ID} is called *decryption component* and it is only used to decrypt messages; the second part K_{ID} is called *delegation component* and its sole purpose is to derive private keys for children nodes. Many HIBE systems in the literature (e.g., [15, 20]) are key-partitioned.

B.1 Security Definitions for HIBE

The standard security notion [37] captures that any coalition of hierarchy entities that are not ancestors of some user should be unable to gain information on messages encrypted for that user.

[5] We use the syntax of [20] which involves an explicit delegation algorithm Derive. Although this algorithm is not explicitly written in [15], it exists as noted in appendix B.2.

In [30], Canetti, Halevi and Katz suggested a weaker security notion, called *selective* security, where the adversary has to choose its target identity upfront.

Definition 7. *[30] A HIBE system with ℓ levels is selectively secure (or* IND-sID-CPA *secure) if no PPT adversary $\mathcal{A}$ has non-negligible advantage in this game:*

1. *The adversary $\mathcal{A}$ chooses a target identity $\mathsf{ID}^\star = (I_1^\star, \ldots, I_{\ell^\star}^\star)$ at depth $\ell^\star < \ell$, for some $\ell^\star$ of its choice. The challenger runs $(mpk, msk) \leftarrow \mathsf{Setup}(\lambda)$ and hands mpk to $\mathcal{A}$.*
2. *$\mathcal{A}$ issues a number of key extraction queries under the rule that no prefix of $\mathsf{ID}^\star$ can be the input of a key extraction query. On input of an identity $\mathsf{ID} = (I_1, \ldots, I_k)$, with $k \leq \ell$, the challenger responds with $d_{\mathsf{ID}} \leftarrow \mathsf{Keygen}(msk, \mathsf{ID})$.*
3. *When $\mathcal{A}$ decides that the first phase is over, it chooses messages m_0, m_1. The challenger flips a coin $d \overset{R}{\leftarrow} \{0,1\}$ and responds with a challenge $C^\star = \mathsf{Encrypt}(mpk, \mathsf{ID}^\star, m_d)$.*
4. *$\mathcal{A}$ issues new queries but cannot ask for the private key of a prefix of $\mathsf{ID}^\star$.*
5. *$\mathcal{A}$ finally outputs a bit $d' \in \{0,1\}$ and wins if $d' = d$. As usual, $\mathcal{A}$'s advantage is quantified as the distance $\mathbf{Adv}^{\mathrm{hibe}}(A) := |\Pr[d' = d] - 1/2|$.*

For our purposes, we need a different and non-standard form of selective security, which mandates that the adversary be unable to maul a private key that it obtained for some target identity $\mathsf{ID}^\dagger$ *even* knowing the master secret key. By "mauling", we mean computing a private key for a different identity $\mathsf{ID}' \neq \mathsf{ID}^\dagger$ but under the same randomness as the received key $d_{\mathsf{ID}^\dagger}$.

Definition 8. *A key-partitioned HIBE system with ℓ levels is said selectively key-robust if no PPT adversary $\mathcal{A}$ has non-negligible advantage in the following game:*

1. *The adversary $\mathcal{A}$ chooses an identity $\mathsf{ID}^\dagger = (I_1^\dagger, \ldots, I_{\ell^\dagger}^\dagger)$ that it wishes to be challenged upon at the depth $\ell^\dagger < \ell$ of its choice. The challenger runs $(mpk, msk) \leftarrow \mathsf{Setup}(\lambda)$ and hands (msk, mpk) to $\mathcal{A}$ along with a challenge consisting of a private key $d_{\mathsf{ID}^\dagger}$ for the identity $\mathsf{ID}^\dagger$.*
2. *$\mathcal{A}$ outputs an identity ID' such that $\mathsf{ID}^\dagger$ is not a prefix[6] of ID' and a decryption component $D_{\mathsf{ID}'}$ (i.e., a private key without delegation component). The adversary wins if: (i) $D_{\mathsf{ID}'}$ is a valid decryption component for ID'; (ii) $D_{\mathsf{ID}'}$ and $d_{\mathsf{ID}^\dagger}$ correspond to the same randomness of the key generation algorithm.*

In Definition 8, we insist that $\mathcal{A}$ is given a full private key for the target identity $\mathsf{ID}^\dagger$ but it only has to output a valid decryption component $D_{\mathsf{ID}'}$ for ID'.

It is also worth insisting that, for the application of this paper, a selective flavor of key-robustness suffices. Indeed, since the number of group members is always polynomial, the target identity $\mathsf{ID}^\dagger$ can be guessed upfront with non-negligible probability in the proof of Lemma 2 (in appendix C).

B.2 The Boneh-Boyen-Goh HIBE

In [15], Boneh, Boyen and Goh (BBG) described the first HIBE scheme where the size of ciphertext does not depend on the depth of the receiver in the hierarchy. The construction bears resemblance with the first selectively secure IBE scheme of Boneh and Boyen [14], which can be seen as a single-level variant of the BBG HIBE. The latter works as follows.

$\mathsf{Setup}(\lambda, \ell)$: given a security parameter $\lambda \in \mathbb{N}$ and the number of levels $\ell \in \mathbb{N}$ in the hierarchy, choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order p, where $p > 2^\lambda$. Choose $\alpha \overset{R}{\leftarrow} \mathbb{Z}_p$, $g, g_2, h_0, h_1, \ldots, h_\ell \overset{R}{\leftarrow} \mathbb{G}$ and compute $g_1 = g^\alpha$. The master public key is defined to be

$$mpk_{\mathsf{BBG}} := \Big((\mathbb{G}, \mathbb{G}_T),\ g,\ g_1,\ g_2,\ \{h_i\}_{i=0}^{\ell}\Big)$$

[6] We assume that hierarchical identities are prefixes of themselves for simplicity.

while the master secret key consists of $msk_{\mathsf{BBG}} := g_2^{\alpha}$. The space of hierarchical identities is $\mathcal{I} = (\mathbb{Z}_p^*)^{\leq \ell}$.

Keygen$\big(msk_{\mathsf{BBG}}, \mathsf{ID} = (I_1, \ldots, I_k)\big)$**:** to generate a private key for $\mathsf{ID} = (I_1, \ldots, I_k) \in (\mathbb{Z}_p^*)^k$ at level k using $\mathsf{msk}_{\mathsf{BBG}} = g_2^{\alpha}$, choose $r \stackrel{R}{\leftarrow} \mathbb{Z}_p$. Then, compute and return

$$d_{\mathsf{ID}} = (D_1, D_2, K_{k+1}, \ldots, K_\ell) = \Big(g_2^{\alpha} \cdot \big(h_0 \cdot \prod_{i=1}^{k} h_i^{I_i}\big)^r,\ g^r,\ h_{k+1}^r, \ldots,\ h_\ell^r\Big) \in \mathbb{G}^{\ell-k+2}. \tag{10}$$

Derive$\big(d_{\mathsf{ID}}, \mathsf{ID}' = (I_1, \ldots, I_k, I_{k+1})\big)$**:** given a private key d_{ID} of the form (10) for the hierarchical identity $\mathsf{ID} = (I_1, \ldots, I_k)$, it is easy to derive a key for the identity $\mathsf{ID}' = (I_1, \ldots, I_k, I_{k+1}) \in (\mathbb{Z}_p^*)^{k+1}$ by choosing $r' \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and computing

$$\begin{aligned} d_{\mathsf{ID}'} &= (D_1', D_2', K_{k+2}', \ldots, K_\ell') \\ &= \Big(D_1 \cdot K_{d+1}^{I_{k+1}} \cdot \big(h_0 \cdot \prod_{i=1}^{k+1} h_i^{I_i}\big)^{r'}, D_2 \cdot g^{r'}, K_{k+2} \cdot h_{k+2}^{r'}, \ldots, K_\ell \cdot h_\ell^{r'}\Big) \qquad (11) \\ &= \Big(g_2^{\alpha} \cdot \big(h_0 \cdot \prod_{i=1}^{k+1} h_i^{I_i}\big)^{r''},\ g^{r''},\ h_{k+2}^{r''}, \ldots,\ h_\ell^{r''}\Big) \in \mathbb{G}^{\ell-k+1}, \end{aligned}$$

where $r'' = r + r'$.

Encrypt$(mpk_{\mathsf{BBG}}, \mathsf{ID} = (I_1, \ldots, I_d), M)$**:** to encrypt $M \in \mathbb{G}$ under $\mathsf{ID} = (I_1, \ldots, I_d) \in (\mathbb{Z}_p^*)^d$, choose $s \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and compute

$$C_0 = M \cdot e(g_1, g_2)^s, \qquad C_1 = g^s, \qquad C_2 = \big(h_0 \cdot h_1^{I_1} \cdots h_d^{I_d}\big)^s$$

The ciphertext is $C = \big(C_0, C_1, C_2\big)$.

Decrypt$(mpk_{\mathsf{BBG}}, d_{\mathsf{ID}}, C)$**:** parse d_{ID} as $(D_1, D_2, K_{d+1}, \ldots, K_\ell) \in \mathbb{G}^{\ell-d+2}$ and the ciphertext C as $\big(C_0, C_1, C_2\big)$. Then, compute and output

$$M = C_0 \cdot e(C_1, D_1)^{-1} \cdot e(C_2, D_2).$$

It is easy to see that this construction is key-partitioned since the private key can be divided into $(D_{\mathsf{ID}}, K_{\mathsf{ID}})$, where $D_{\mathsf{ID}} = (D_1, D_2) \in \mathbb{G}^2$ is only used to decrypt and $K_{\mathsf{ID}} = (K_{k+1}, \ldots, K_\ell) \in \mathbb{G}^{\ell-k+2}$ is only useful for delegations.

When the private key for ID' is derived from the private key for ID, the randomizer $r' \in \mathbb{Z}_p$ in (11) allows making sure that derived private keys are indistinguishable from original keys that are generated directly at level $k+1$.

In our application, we will require that key be derived without any randomization. Namely, a private key for ID' is always derived as per

$$\begin{aligned} d_{\mathsf{ID}'} = (D_1', D_2', K_{k+2}', \ldots, K_\ell') &= \Big(D_1 \cdot K_{d+1}^{I_{k+1}},\ D_2,\ K_{k+2}, \ldots,\ K_\ell\Big) \\ &= \Big(g_2^{\alpha} \cdot \big(h_0 \cdot \prod_{i=1}^{k+1} h_i^{I_i}\big)^r,\ g^r,\ h_{k+2}^r, \ldots,\ h_\ell^r\Big) \in \mathbb{G}^{\ell-k+1}. \end{aligned}$$

For this reason, a private key and its descendants will always share the same component D_2. However, it does not affect the security of the group signature since, in the join protocol, users are always given freshly generated HIBE private keys.

The following lemma demonstrates that the BBG HIBE is selectively key-robust under the Diffie-Hellman assumption. The proof implicitly relies on the fact (implicitly noted in [43, 42]) that BLS-type signatures [17] can be proved secure in the standard model when the number of signing queries is bounded by a small constant (such as one here since the one-time public key g^r is used as a one-time public key).

Lemma 1. *The BBG HIBE scheme is selectively key-robust assuming that the CDH assumption holds in $\mathbb{G}$. More precisely, a selective key-robustness adversary $\mathcal{A}$ with advantage ε implies an algorithm $\mathcal{B}$ solving the CDH problem with advantage $\varepsilon \cdot (1 - 1/p)$.*

Proof. Towards a contradiction, let us assume that a selective adversary $\mathcal{A}$ has non-negligible advantage in the game of Definition 8. We show that $\mathcal{A}$ allows breaking the CDH assumption.

Algorithm $\mathcal{B}$ receives as input a CDH instance $(g, g^a, g^b) \in \mathbb{G}^3$ and undertakes to compute g^{ab}. At the beginning of its interaction with $\mathcal{A}$, the latter chooses a target identity $\mathsf{ID}^\dagger = (I_1^\dagger, \ldots, I_{\ell^\dagger}^\dagger) \in (\mathbb{Z}_p^*)^{\ell^\dagger}$ at the depth $\ell^\dagger \leq \ell$ of its choice. Then, $\mathcal{B}$ generates the master key pair $(msk_{\mathsf{BBG}}, mpk_{\mathsf{BBG}})$ by choosing $\alpha \stackrel{R}{\leftarrow} \mathbb{Z}_p$, $g_2 \stackrel{R}{\leftarrow} \mathbb{G}$ and setting $g_1 = g^\alpha$ as in the normal setup algorithm. Then, it picks $\gamma_1, \ldots, \gamma_{\ell^\dagger} \stackrel{R}{\leftarrow} \mathbb{Z}_p$, $\delta_0, \ldots, \delta_\ell \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and defines

$$\begin{aligned} h_0 &= g^{\delta_0} \cdot (g^b)^{-\sum_{i=1}^{\ell^\dagger} \gamma_i I_i^\dagger} \\ h_i &= g^{\delta_i} \cdot (g^b)^{\gamma_i} && \text{for } i = 1, \ldots, \ell^\dagger \\ h_i &= g^{\delta_i} && \text{for } i = \ell^\dagger + 1, \ldots, \ell. \end{aligned}$$

To generate a private key $d_{\mathsf{ID}^\dagger} = (D_1, D_2, K_{\ell^\dagger+1}, \ldots, K_\ell)$ for the target identity $\mathsf{ID}^\dagger$, $\mathcal{B}$ sets

$$\begin{aligned} D_1 &= g_2^\alpha \cdot (g^a)^{\delta_0 + \sum_{i=1}^{\ell^\dagger} \delta_i I_i^\dagger} \\ D_2 &= g^a \\ K_i &= (g^a)^{\delta_i} && \text{for } i = \ell^\dagger + 1, \ldots, \ell, \end{aligned}$$

which form a valid private key for the random exponent $r = a$.

The adversary $\mathcal{A}$ is given $(mpk_{\mathsf{BBG}}, msk_{\mathsf{BBG}} = g_2^\alpha)$ and the private key $d_{\mathsf{ID}^\dagger}$. Its goal will be to produce a valid decryption component $D_{\mathsf{ID}'} = (D_1', D_2')$ corresponding to an identity $\mathsf{ID}' = (I_1', \ldots, I_k') \in (\mathbb{Z}_p^*)^k$, for some $k \in \{1, \ldots, \ell\}$, that $\mathsf{ID}^\dagger$ is not a prefix of. In addition, $D_{\mathsf{ID}'}$ should correspond to the same random exponent $r = a$ as $d_{\mathsf{ID}^\dagger}$ (in other words, $D_1' = D_2 = g^a$).

When $\mathcal{A}$ outputs its result $(\mathsf{ID}', D_{\mathsf{ID}'})$, we distinguish the following situations.

- If $k \leq \ell^\dagger$, we have

$$h_0 \cdot \prod_{i=1}^{k} h_i^{I_i'} = g^{\delta_0 + \sum_{i=1}^k \delta_i I_i'} \cdot (g^b)^{\sum_{i=1}^k \gamma_i \cdot (I_i' - I_i^\dagger) - \sum_{i=k+1}^{\ell^\dagger} \gamma_i I_i^\dagger} \tag{12}$$

  and, with overwhelming probability $1 - 1/p$, it holds that

$$\sum_{i=1}^{k} \gamma_i \cdot (I_i' - I_i^\dagger) - \sum_{i=k+1}^{\ell^\dagger} \gamma_i I_i^\dagger \neq 0. \tag{13}$$

  Indeed, the vector $\vec{\gamma} = (\gamma_1, \ldots, \gamma_{\ell^\dagger})$ is chosen uniformly in $\mathbb{Z}_p^{\ell^\dagger}$ and it is independent of $\mathcal{A}$'s view.

  - If $k = \ell^\dagger$, we must have $I_i' \neq I_i^\dagger$ for at least one $i \in \{1, \ldots, \ell^\dagger\}$. Since the coordinates of $\vec{\gamma}$ are independent and uniformly distributed, the probability to have $\sum_{i=1}^{\ell^\dagger} \gamma_i \cdot (I_i' - I_i^\dagger) = 0$ is at most $1/p$ since we are bounding the probability of a random vector $\vec{\gamma}$ to be orthogonal to a given non-zero vector of $\mathbb{Z}_p^{\ell^\dagger}$.
  - If $k < \ell^\dagger$, we may have $I_i' = I_i^\dagger$ for each $i \in \{1, \ldots, k\}$ (*i.e.*, ID' may be a prefix of $\mathsf{ID}^\dagger$). In this situation, the probability to have $\sum_{i=k+1}^{\ell^\dagger} \gamma_i I_i^\dagger = 0$ is also $1/p$ since $(\gamma_{k+1}, \ldots, \gamma_{\ell^\dagger})$ is independent of $\mathcal{A}$'s view and identities $I_{k+1}^\dagger, \ldots, I_{\ell^\dagger}^\dagger$ are always non-zero. Finally, if ID' is not a prefix of $\mathsf{ID}^\dagger$, there exists $i \in \{1, \ldots, k\}$ such that $I_i' \neq I_i^\dagger$. Then, the same argument as in previous cases applies.

Since $\mathcal{A}$ presumably outputs a decryption component $D_{\mathsf{ID}'} = (D_1, D_2) = \big(g_2^{\alpha} \cdot (h_0 \cdot \prod_{i=1}^{k} h_i^{I'_i})^a,\ g^a\big)$ with non-negligible probability ε, $\mathcal{B}$ can compute

$$g^{ab} = \Big(\frac{D_1}{g_2^{\alpha} \cdot (g^a)^{\delta_0 + \sum_{i=1}^{k} \delta_i I'_i}}\Big)^{1/(\sum_{i=1}^{k} \gamma_i \cdot (I'_i - I_i^{\dagger}) - \sum_{i=k+1}^{\ell^{\dagger}} \gamma_i I_i^{\dagger})}$$

with probability $\varepsilon \cdot (1 - 1/p)$.

- If $k > \ell^{\dagger}$, there exists $i \in \{1, \ldots, \ell^{\dagger}\}$ such that $I'_i \neq I_i^{\dagger}$ since $\mathsf{ID}^{\dagger}$ cannot be a prefix of ID'. In this case, we can write

$$h_0 \cdot \prod_{i=1}^{k} h_i^{I'_i} = g^{\delta_0 + \sum_{i=1}^{k} \delta_i I'_i} \cdot (g^b)^{\sum_{i=1}^{\ell^{\dagger}} \gamma_i \cdot (I'_i - I_i^{\dagger})}$$

where $\sum_{i=1}^{\ell^{\dagger}} \gamma_i \cdot (I'_i - I_i^{\dagger}) \neq 0$ with probability at least $1 - 1/p$. Then, the CDH solution g^{ab} can be found in the same way as in the case $k \leq \ell^{\dagger}$.

□

C Deferred Lemma for the Security against Misidentification Attacks

Lemma 2. *The advantage of any Type II.b forger $\mathcal{A}$ is at most*

$$\mathbf{Adv}_{\mathcal{A}}^{\text{mis-id-II.b}}(\lambda) \leq 4 \cdot N^2 \cdot \Big(1 - \frac{1}{p}\Big) \cdot \mathbf{Adv}^{\text{CDH}}(\lambda)$$

where N denotes the maximal number of users.

Proof. At the beginning of its interaction with its challenger, our selective key-robustness adversary $\mathcal{B}$ chooses a random node $x_j \in \mathsf{T}$ and a random descendant x'_j of x_j (alternatively, $\mathcal{B}$ can more simply choose two distinct random nodes in the tree and, with some probability, x'_j will be in the subtree rooted at x_j). Since x'_j is a descendant of x_j, its label $\langle x'_j \rangle$ can be written $\langle x'_j \rangle = \langle x_j \rangle || w_{\ell_1} \ldots w_{\ell_2}$, for some integers $\ell_1, \ell_2 \in \{1, \ldots, \ell\}$ and where $w_i \in \{0, 1\}$ for each $i \in \{\ell_1, \ldots, \ell_2\}$. Then, $\mathcal{B}$ declares $\mathsf{ID}^{\dagger} = (\mathcal{H}(\langle x_j \rangle), \mathcal{H}(w_{\ell_1}), \ldots, \mathcal{H}(w_{\ell_2}))$ as its target identity at level $\ell_2 - \ell_1 + 2$. The key-robustness challenger replies by returning a master key pair $(msk_{\mathsf{BBG}}, mpk_{\mathsf{BBG}})$ consisting of

$$mpk_{\mathsf{BBG}} = \Big((\mathbb{G}, \mathbb{G}_T),\ g,\ g_1 = g^{\alpha},\ g_2,\ \{h_i\}_{i=0}^{\ell}\Big), \qquad msk_{\mathsf{BBG}} = g_2^{\alpha}$$

together with a private key $d_{\mathsf{ID}^{\dagger}} = (D_1^{\dagger}, D_2^{\dagger}, K_{\ell_2 - \ell_1 + 3}^{\dagger}, \ldots, K_{\ell}^{\dagger})$ for the identity $\mathsf{ID}^{\dagger}$.

Then, $\mathcal{B}$ uses mpk_{BBG} to construct the group public key $\mathcal{Y}$ and generates all other public key elements (including $pk_{\mathsf{AHO}}^{(0)}$ and $pk_{\mathsf{AHO}}^{(1)}$) according to the specification of the setup algorithm. In particular, $\mathcal{B}$ retains the group manager's secret key $\mathcal{S}_{\mathsf{GM}} = (sk_{\mathsf{AHO}}^{(0)}, sk_{\mathsf{AHO}}^{(1)})$ and uses it to answer $Q_{\mathsf{a\text{-}join}}$-queries.

At each $Q_{\mathsf{a\text{-}join}}$-query, $\mathcal{B}$ executes the join protocol on behalf of $\mathsf{J_{GM}}$ and proceeds exactly as the real $\mathsf{J_{GM}}$ does (recall that it knows $\mathcal{S}_{\mathsf{GM}}$ and can thus perfectly simulate $\mathsf{J_{GM}}$) with one exception. Namely, when executing step b.1 of Join, if the private key D_w has to be computed for the target identity $\mathsf{ID}^{\dagger}$, $\mathcal{B}$ uses the private key $d_{\mathsf{ID}^{\dagger}}$ that it received from its challenger to compute

$$(D_{w,1}, D_{w,2}, K_{w,\ell_2 - \ell_1 + 3}, \ldots, K_{w,\ell}) = \big(D_1^{\dagger}/g_2^{\alpha},\ D_2^{\dagger},\ K_{\ell_2 - \ell_1 + 3}^{\dagger}, \ldots,\ K_{\ell}^{\dagger}\big).$$

As for Q_{pub}, Q_{revoke}, Q_{read} and Q_{OA}-queries, $\mathcal{B}$ simply answers them as the real oracles would.

If the game terminates and $\mathcal{B}$ did not have to compute a private key for $\mathsf{ID}^{\dagger}$ at some $Q_{\mathsf{a\text{-}join}}$-query, then

$\mathcal{B}$ halts and reports failure since it must have guessed the wrong tree nodes x_j and x'_j. Otherwise (*i.e.*, if $d_{\mathsf{ID}^\dagger}$ was used to compute part of a membership certificate), we know that $\mathcal{A}$'s forgery $\sigma^\star$ will contain a committed HIBE private key $(D_1^\star, D_2^\star)$ and a committed value $X^\star$ such that $(X^\star, D_2^\star)$ is one of the pairs that were signed by $\mathcal{B}$ during some $Q_{\mathsf{a\text{-}join}}$-query. Moreover, the pair $(X^\star, D_2^\star)$ was associated with some HIBE private key $d_{\mathsf{ID}^\diamond} = (D_1, D_2^\star, K_{\ell_2-\ell_1+3}, \ldots, K_\ell)$ for a certain hierarchical identity $\mathsf{ID}^\diamond$ at step b.1 of Join. Since that identity is entirely defined by two nodes at the extremities of a path in the tree T, these two nodes happen to be x_j and x'_j – so that $\mathsf{ID}^\diamond = \mathsf{ID}^\dagger$ – with non-negligible probability $1/(2N-1)^2 > 1/4N^2$.

Therefore, with probability at least $1/4N^2$, the value $D_2^\star$ is precisely the element $D_2^\dagger$ of the challenge private key $d_{\mathsf{ID}^\dagger}$ sent by the key-robustness challenger. Also, we note that $d_{\mathsf{ID}^\diamond}$ and $(D_1^\star, D_2^\star)$ necessarily correspond to distinct identities as the signature $\sigma^\star$ would not trace to a revoked user otherwise. If the desirable event $\mathsf{ID}^\diamond = \mathsf{ID}^\dagger$ comes about, this implies that either:

- $(D_1^\star, D_2^\star) = (D_1^\dagger, D_2^\dagger)$, which means that $d_{\mathsf{ID}^\dagger}$ and $(D_1^\star, D_2^\star)$ correspond to distinct hierarchical identities $\mathsf{ID}^\dagger = (I_1, \ldots, I_{\ell_2-\ell_1+2})$ and $(I'_1, \ldots, I_k)$, with $k \in \{1, \ldots, \ell\}$, such that

  $$h_0 \cdot \prod_{i=1}^{\ell_2-\ell_1+2} h_i^{I_i} = h_0 \cdot \prod_{i=1}^{k'} h_i^{I'_i}.$$

  Such a collision is known (as shown in [43][Section 1.2], for example) to occur with negligible probability under the discrete logarithm assumption.
- $D_2^\star = D_2^\dagger$ and $D_1^\star \neq D_1^\dagger$, in which case $\mathcal{B}$ wins the selective key-robustness game. It does so by outputting the decryption component $(g_2^\alpha \cdot D_1^\star, D_2^\star)$ – after having extracted $(D_1^\star, D_2^\star)$ from $\{com_{D_i}^\star\}_{i=1}^2$ using (β_1, β_2) – and the identity ID' corresponding to the HIBE ciphertext $C_l^\star$ of the revocation list. Indeed, if $\mathcal{B}$ correctly guessed x_j and x'_j, ID' cannot be a descendant of $\mathsf{ID}^\dagger$ as long as $\sigma^\star$ opens to a revoked user in $U^a \cap \mathcal{R}_{t^\star}$.

Since the probability to have $\mathsf{ID}^\diamond = \mathsf{ID}^\dagger$ is at least $1/4N^2$ and due to the multiplicative factor $(1-1/p)$ in the statement of Lemma 1, the announced result follows. □

D Security against Framing Attacks and Anonymity

D.1 Framing Attacks

Theorem 2 (Non-frameability). *The scheme is secure against framing attacks assuming that: (i) the q_b-SDH assumption holds in $\mathbb{G}$, where q_b is the maximal number of $Q_{\mathsf{b\text{-}join}}$-queries; (ii) Σ is a strongly unforgeable one-time signature.*

Proof. As in [39], we consider two kinds of framing attacks that can be possibly mounted by a non-frameability adversary $\mathcal{A}$.

- **Type I attacks**: the adversary $\mathcal{A}$ generates a forgery $\sigma^\star = \big(\mathsf{VK}^\star, \Psi_1^\star, \Psi_2^\star, \Psi_3^\star, \Psi_4^\star, \Psi_5^\star, \Omega^\star, \mathbf{com}^\star, \mathbf{\Pi}^\star, \sigma_{ots}^\star\big)$ for which the one-time verification key $\mathsf{VK}^\star$ was used by some honest group member $i \in U^b$ when answering a Q_{sig}-query.
- **Type II attacks**: $\mathcal{A}$ outputs a forgery $\sigma^\star = \big(\mathsf{VK}^\star, \Psi_1^\star, \Psi_2^\star, \Psi_3^\star, \Psi_4^\star, \Psi_5^\star, \Omega^\star, \mathbf{com}^\star, \mathbf{\Pi}^\star, \sigma_{ots}^\star\big)$ for which the one-time verification key $\mathsf{VK}^\star$ was never used by Q_{sig} to answer a signing query on behalf of an honest user $i \in U^b$.

It is immediate that Type I attacks imply a breach in the unforgeability of the one-time signature. Lemma 3 shows that no PPT adversary can produce a Type II forgery as long as the Strong Diffie-Hellman assumption holds. □

Lemma 3. *The scheme is secure against framing attacks of Type II if the q_s-SDH problem is hard. More precisely, the advantage of any adversary after q_s Q_{sig}-queries and q_b $Q_{\mathsf{b\text{-}join}}$-queries is at most* $\mathbf{Adv}^{\text{fra-II}}(\lambda) \leq q_b \cdot \mathbf{Adv}^{\text{q}_\text{s}\text{-SDH}}(\lambda)$.

Proof. By hypothesis, the adversary $\mathcal{A}$ comes up with a forgery $(M^\star, \sigma^\star)$ that opens to some honest user $i \in U^b$ and that did not issue a signature containing the verification key $\mathsf{VK}^\star$. The same proof as in [39] shows that the Strong Diffie-Hellman assumption can be broken.

Given a problem instance $(\tilde{g}, \tilde{g}^a, \ldots, \tilde{g}^{(a^{q_s})}) \in \mathbb{G}^{q_s+1}$, the simulator $\mathcal{B}$ generates q_s one-time signature keys pairs $(\mathsf{SK}_i, \mathsf{VK}_i) \leftarrow \mathcal{G}(\lambda)$ for $i = 1$ to q_s. Then, using standard techniques (see [13][Lemma 3.2]) it builds a generator g and a randomly distributed public value $X^\dagger = g^a$ – which implicitly defines $x^\dagger = \log_g(X^\dagger) = a$ – such that it knows $\{(g^{1/(a+\mathsf{VK}_i)}, \mathsf{VK}_i)\}_{i=1}^{q_s}$.

Next, using the newly generated g, $\mathcal{B}$ generates key pairs $\{(sk_{\mathsf{AHO}}^{(b)}, pk_{\mathsf{AHO}}^{(b)})\}_{b=0,1}$ for the AHO signature (note that group elements of $\{pk_{\mathsf{AHO}}^{(b)}\}_{b=0,1}$ are computed as powers of g) and uses $pk_{\mathsf{AHO}}^{(0)}, pk_{\mathsf{AHO}}^{(1)}$ to form the group public key

$$\mathcal{Y} = \big(g,\ pk_{\mathsf{AHO}}^{(0)},\ pk_{\mathsf{AHO}}^{(1)},\ mpk_{\mathsf{BBG}},\ \mathbf{f},\ U,\ V,\ \mathcal{H},\ \Sigma\big).$$

In the latter, the Groth-Sahai CRS $\mathbf{f} = (\vec{f_1}, \vec{f_2}, \vec{f_3})$ is prepared for the perfect soundness setting, *i.e.*, with $\vec{f_1} = (f_1 = g^{\beta_1}, 1, g)$, $\vec{f_2} = (1, f_2 = g^{\beta_2}, g)$ and $\vec{f_3} = \vec{f_1}^{\xi_1} \odot \vec{f_2}^{\xi_2}$, where $\xi_1, \xi_2 \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$.

Should the adversary $\mathcal{A}$ decide to corrupt the group manager or the opening authority during the game, $\mathcal{B}$ has $\mathcal{S}_{\mathsf{GM}} = (sk_{\mathsf{AHO}}^{(0)}, sk_{\mathsf{AHO}}^{(1)})$ and $\mathcal{S}_{\mathsf{OA}} = (\beta_1, \beta_2) = (\log_g(f_1), \log_g(f_2))$ at its disposal. At the beginning of the game, $\mathcal{B}$ picks a random index $j^\star \stackrel{R}{\leftarrow} \{1, \ldots, q_b\}$ and interacts with $\mathcal{A}$ as follows.

- Q_{keyGM}-queries: if $\mathcal{A}$ decides to corrupt the group manager, $\mathcal{B}$ surrenders $\mathcal{S}_{\mathsf{GM}} = (sk_{\mathsf{AHO}}^{(0)}, sk_{\mathsf{AHO}}^{(1)})$.
- $Q_{\mathsf{b\text{-}join}}$-queries: when $\mathcal{A}$, acting as a rogue group manager, requests the introduction of a new honest user i in the group, $\mathcal{B}$ starts interacting with $\mathcal{A}$ in an execution of Join and runs $\mathsf{J_{user}}$ on behalf of the prospective user. Namely, $\mathcal{B}$'s behavior depends on the index $j \in \{1, \ldots, q_b\}$ of the $Q_{\mathsf{b\text{-}join}}$-query.
  - If $j \neq j^\star$, $\mathcal{B}$ follows exactly the specification of $\mathsf{J_{user}}$.
  - If $j = j^\star$, $\mathcal{B}$ sends the value $X^\dagger$ to $\mathsf{J_{GM}}$ at step 1 of Join. This implicitly defines user $j^\star$'s membership secret to be the unknown exponent $\mathsf{sec}_{j^\star} = a$ of the SDH instance. In subsequent steps of the join protocol, $\mathcal{B}$ proceeds as the real $\mathsf{J_{user}}$ would. When Join terminates, $\mathcal{B}$ obtains a membership certificate $\mathsf{cert}_{j^\star} = \big(\langle v_j \rangle, \{\{d_w, \sigma_w\}_{w \in \mathsf{copath}_{x_j}}\}_{j=0}^{\ell}, X^\dagger\big)$.
- Q_{pub}-queries: can be treated as in the real game, by having the simulator return $\mathcal{Y}$.
- Q_{sig}-queries: when $\mathcal{A}$ asks user $i \in U^b$ to sign a message M, the simulator $\mathcal{B}$ can answer the query by running the real signature generation algorithm if $i \neq j^\star$. Otherwise (namely, if $i = j^\star$), $\mathcal{B}$ uses the next available pair $\{(g^{1/(a+\mathsf{VK}_i)}, \mathsf{VK}_i)\}_{i=1}^{q_s}$ to define σ_{VK_i}. It also recalls the membership certificate $\mathsf{cert}_{j^\star} = \big(\langle v_j \rangle, \{\{d_w, \sigma_w\}_{w \in \mathsf{copath}_{x_j}}\}_{j=0}^{\ell}, X^\dagger\big)$ that it obtained from the $\mathsf{J_{GM}}$-executing adversary at the $j^\star$-th $Q_{\mathsf{b\text{-}join}}$-query. It is easy to see that, using σ_{VK_i} and $\mathsf{cert}_{j^\star}$, it can easily generate all signature components and sign them all using SK_i.

Finally, $\mathcal{A}$ outputs a signature $\sigma^\star = \big(\mathsf{VK}^\star, \Psi_1^\star, \Psi_2^\star, \Psi_3^\star, \Psi_4^\star, \Psi_5^\star, \Omega^\star, \mathbf{com}^\star, \mathbf{\Pi}^\star, \sigma_{ots}^\star\big)$, for some message $M^\star$, that opens to some user $i^\star \in U^b$ who did not sign $M^\star$. At this point, $\mathcal{B}$ halts and declares failure if it turns out that $X^\dagger \neq \Psi_3^\star \cdot \Psi_1^{\star -1/\beta_1} \cdot \Psi_2^{\star -1/\beta_2}$ since, in this case, it was unfortunate when drawing the random index $j^\star$. Still, with probability $1/q_b$, the signature $\sigma^\star$ opens to the user introduced at the $j^\star$-th $Q_{\mathsf{b\text{-}join}}$-query and $(\Psi_1^\star, \Psi_2^\star, \Psi_3^\star)$ does decrypt to $X^\star$. In this situation, the perfect soundness of the proof system ensures that $com_{\sigma_{\mathsf{VK}^\star}}^\star$ is a commitment to a group element $\sigma_{\mathsf{VK}^\star}^\star$ such that $e(\sigma_{\mathsf{VK}^\star}^\star, X^\dagger \cdot g^{\mathsf{VK}^\star}) = e(g, g)$. Since $\sigma^\star$ is a Type II forgery, $\mathcal{B}$ can use β_1, β_2 to compute a BBS decryption of $com_{\sigma_{\mathsf{VK}^\star}}^\star$ and obtain a solution $(\sigma_{\mathsf{VK}^\star}, \mathsf{VK}^\star)$ to the q_b-SDH instance. □

D.2 Anonymity

As for the anonymity property, it naturally relies on the DLIN assumption. The proof is essentially identical to that of Lemma 5 in [39] but we give it for completeness.

Theorem 3 (Anonymity). *The advantage of any anonymity adversary is at most*

$$\mathbf{Adv}^{\mathrm{anon}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{ots}}(\lambda) + 3 \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\lambda),$$

where the first term is $\mathcal{A}$'s probability of breaking the strong unforgeability of the one-time signature.

Proof. We consider a sequence of games at the end of which even an unbounded adversary has no advantage. In Game i, we call S_i the event that $\mathcal{A}$ wins and define $Adv_i = |\Pr[S_i] - 1/2|$.

Game 1: is the experiment of definition 6. In the play stage, the adversary $\mathcal{A}$ can obtain the group public key $\mathcal{Y}$, the group manager's private key $\mathcal{S}_{\mathsf{GM}} = (sk_{\mathsf{AHO}}^{(0)}, sk_{\mathsf{AHO}}^{(1)})$. It can also ask for the opening of any group signature and read/write the content of $\mathsf{state}_{\mathcal{I}}$. When it decides to enter the challenge phase, it outputs a message $M^\star$, a period index $t^\star$ and two membership certificate/secret $(\mathsf{cert}_0^\star, \mathsf{sec}_0^\star)$ and $(\mathsf{cert}_1^\star, \mathsf{sec}_1^\star)$ such that $\mathsf{cert}_b^\star \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_b^\star$ for $b = 0, 1$. The simulator $\mathcal{B}$ flips a fair coin $d \xleftarrow{R} \{0,1\}$ and computes $\sigma^\star \leftarrow \mathsf{Sign}(\mathcal{Y}, t^\star, RL_{t^\star}, \mathsf{cert}_d^\star, \mathsf{sec}_d^\star, M^\star)$, where $t^\star$ is determined by the history of Q_{revoke}-queries. The signature $\sigma^\star$ is given as a challenge to $\mathcal{A}$ who has to guess $d \in \{0,1\}$ after another series of queries (under the natural restriction of not querying the opening of $\sigma^\star$). We have $Adv_1 = \mathbf{Adv}^{\mathrm{anon}}(\mathcal{A})$.

Game 2: is as **Game** 1 but $\mathcal{B}$ halts if $\mathcal{A}$ queries the opening of a signature σ containing the same one-time verification key $\mathsf{VK}^\star$ as in the challenge phase (we assume w.l.o.g. that $(\mathsf{SK}^\star, \mathsf{VK}^\star)$ is generated at the outset of the game). If such a query is made before the challenge phase, it means that $\mathcal{A}$ was able to forge a one-time signature even without having seen a signature. If the query occurs after the challenge phase, then the strong unforgeability of Σ is broken. We can thus write $|\Pr[S_2] - \Pr[S_1]| \leq \mathbf{Adv}^{\mathrm{ots}}(\lambda)$.

Game 3: we change the generation of $\mathcal{Y}$ so as to answer Q_{open}-queries without using the secret exponents $\beta_1, \beta_2 \in \mathbb{Z}_p$ that define $\mathcal{S}_{\mathsf{OA}}$. To this end, $\mathcal{B}$ chooses $\alpha_u, \alpha_v \xleftarrow{R} \mathbb{Z}_p^*$, and defines $U = g^{-\mathsf{VK}^\star} \cdot f_1^{\alpha_u}$, and $V = g^{-\mathsf{VK}^\star} \cdot f_2^{\alpha_v}$. It is not hard to see (see [47] for details) that, for any Q_{open}-query containing a BBS encryption $(\Psi_1, \Psi_2, \Psi_3) = (f_1^{z_1}, f_2^{z_2}, X \cdot g^{z_1+z_2})$, the values (Ψ_4, Ψ_5) reveal g^{z_1} and g^{z_2} (and thus the encrypted X) since $\mathsf{VK} \neq \mathsf{VK}^\star$ unless the event introduced in Game 2 occurs. To generate the challenge signature $\sigma^\star$ at epoch $t^\star$, $\mathcal{B}$ first computes $(\Psi_1^\star, \Psi_2^\star, \Psi_3^\star)$ and then $(\Psi_4^\star, \Psi_5^\star) = (\Psi_1^{\star\alpha_u}, \Psi_2^{\star\alpha_v})$. It sets the challenge signature to be $\sigma^\star = (\mathsf{VK}^\star, \Psi_1^\star, \Psi_2^\star, \Psi_3^\star, \Psi_4^\star, \Psi_5^\star, \Omega^\star, \mathbf{com}^\star, \mathbf{\Pi}^\star, \sigma_{ots}^\star)$. It can be checked that the distributions of $\mathcal{Y}$ and $\sigma^\star$ are unchanged and we have $\Pr[S_3] = \Pr[S_2]$.

Game 4: in the setup phase, we generate the CRS $\mathbf{f} = (\vec{f_1}, \vec{f_2}, \vec{f_3})$ of the proof system for the perfect WI setting. We choose $\vec{f_3} = \vec{f_1}^{\xi_1} \odot \vec{f_2}^{\xi_2} \odot (1,1,g)^{-1}$ instead of $\vec{f_3} = \vec{f_1}^{\xi_1} \odot \vec{f_2}^{\xi_2}$ so that $\vec{f_1}$, $\vec{f_2}$ and $\vec{f_3}$ are linearly independent. Any significant change in $\mathcal{A}$'s behavior yields a distinguisher for the DLIN problem and we can write $|\Pr[S_4] - \Pr[S_3]| = 2 \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\mathcal{B})$. As noted in [40], proofs in the WI setting reveal no information on which witnesses they were generated from.

Game 5: we modify the generation of the challenge $\sigma^\star$ and use the trapdoor of the CRS (*i.e.*, ξ_1, ξ_2 s.t. $\vec{\varphi} = \vec{f_1}^{\xi_1} \odot \vec{f_2}^{\xi_2}$) to simulate proofs $\{\pi_{eq\text{-}com,j}\}_{j=1}^3$ that $(\Psi_1^\star, \Psi_2^\star, \Psi_3^\star)$ and com_X encrypt of the same value. It is known [40] that linear multi-exponentiation equations always have perfectly NIZK proofs on a simulated CRS. For, any satisfiable relation, (ξ_1, ξ_2) allows generating proofs without using the witnesses τ_1, τ_2, τ_3 for which (9) holds and simulated proofs are perfectly indistinguishable from real ones. Hence, $\Pr[S_5] = \Pr[S_4]$.

Game 6: in the computation of $\Psi_3^\star$, we now replace $g^{z_1+z_2}$ by a random group element in the challenge $\sigma^\star$. Since $\mathcal{B}$ does not explicitly use $z_1 = \log_{f_1}(\Psi_1^\star)$, $z_2 = \log_{f_2}(\Psi_2^\star)$, any change in $\mathcal{A}$'s behavior yields a distinguisher for the DLIN problem and $|\Pr[S_6] - \Pr[S_5]| \leq \mathbf{Adv}^{\mathrm{DLIN}}(\mathcal{B})$. In Game 6, we have $\Pr[S_6] = 1/2$. Indeed, when we consider the challenge $\sigma^\star$, Groth-Sahai commitments are all perfectly hiding in the WI setting and proofs $\mathbf{\Pi}$ reveal nothing about the underlying witnesses (in particular, NIZK proofs $\{\pi_{eq\text{-}com,j}\}_{j=1}^{3}$ are generated without using them) and $(\Psi_1^\star, \Psi_2^\star, \Psi_3^\star)$ perfectly hides $X^\star$. Finally, randomized signature components $\Omega^\star = \{{\Theta'_{l,i}}^\star, {\theta'_{l,i}}^\star\}_{i\in\{3,4,6,7\}}$ are information-theoretically independent of the corresponding messages and the remaining components of AHO signatures $\Theta_l^\star$ and $\theta_l^\star$.

When combining the above, $\mathcal{A}$'s advantage can be bounded by $\mathbf{Adv}^{\mathrm{anon}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{ots}}(\lambda) + 3\cdot\mathbf{Adv}^{\mathrm{DLIN}}(\lambda)$ as stated by the theorem. □

E A Construction Based on the Complete Subtree Method

The following construction uses the public-key variant (suggested in [51, 34]) of the CS method, which does not require a hierarchical IBE: a single-level selectively secure IBE scheme such as the one described by Boneh and Boyen [14] suffices. As in our construction based on the SD method, we do not need to use the master secret key of the IBE system.

In the upcoming description, the main difference with the scheme of section 3 is the way to distribute IBE private keys in the join protocol. Other algorithms are essentially unchanged.

As in section 3, the number of users is assumed to be $N = 2^\ell$ so that each group member is assigned to a leaf of the tree. Again, each node is assigned a unique identifier. For simplicity, we define the identifier $\mathsf{ID}(x) \in \{1, \ldots, 2N-1\}$ of node x to be $\mathsf{ID}(x) = 2 \cdot \mathsf{ID}(\mathsf{parent}(x)) + b$, where $\mathsf{parent}(x)$ denotes x's father in the tree and $b = 0$ (resp. $b = 1$) if x is the left (resp. right) child of its father. The root of the tree is assigned the identifier $\mathsf{ID}_\epsilon = 1$.

Setup(λ, N)**:** given a security parameter $\lambda \in \mathbb{N}$ and the permitted number of users $N = 2^\ell$,

1. Choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$, with a generator $g \stackrel{R}{\leftarrow} \mathbb{G}$.
2. Generate two key pairs $(sk^{(0)}_{\mathsf{AHO}}, pk^{(0)}_{\mathsf{AHO}})$ and $(sk^{(1)}_{\mathsf{AHO}}, pk^{(1)}_{\mathsf{AHO}})$ for the AHO signature in order to sign messages of two group elements. These key pairs consist of
$$pk^{(d)}_{\mathsf{AHO}} = \Big(G_r^{(d)},\ H_r^{(d)},\ G_z^{(d)} = G_r^{\gamma_z^{(d)}},\ H_z^{(d)} = H_r^{\delta_z^{(d)}},\ \{G_i^{(d)} = G_r^{\gamma_i^{(d)}}, H_i^{(d)} = H_r^{\delta_i^{(d)}}\}_{i=1}^{2},\ A^{(d)},\ B^{(d)}\Big)$$
and $sk^{(d)}_{\mathsf{AHO}} = \big(\alpha_a^{(d)}, \alpha_b^{(d)}, \gamma_z^{(d)}, \delta_z^{(d)}, \{\gamma_i^{(d)}, \delta_i^{(d)}\}_{i=1}^{2}\big)$, where $d \in \{0,1\}$.
3. As a CRS for the NIWI proof system, select vectors $\mathbf{f} = (\vec{f_1}, \vec{f_2}, \vec{f_3})$ s.t. $\vec{f_1} = (f_1, 1, g) \in \mathbb{G}^3$, $\vec{f_2} = (1, f_2, g) \in \mathbb{G}^3$, and $\vec{f_3} = \vec{f_1}^{\,\xi_1} \cdot \vec{f_2}^{\,\xi_2}$, with $f_1 = g^{\beta_1}, f_2 = g^{\beta_2} \stackrel{R}{\leftarrow} \mathbb{G}$ and $\beta_1, \beta_2, \xi_1, \xi_2 \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$.
4. Choose $(U, V) \stackrel{R}{\leftarrow} \mathbb{G}^2$ that, together with f_1, f_2, g, will form a public key for an IND-CCA2 cryptosystem.
5. Generate a master public key mpk_{BB} for the Boneh-Boyen IBE. Such a public key consists of $mpk_{\mathsf{BB}} = \big(h_0, h_1\big)$ and, again, no master secret key is needed.
6. Select a strongly unforgeable one-time signature $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$.
7. Set $\mathcal{S}_{\mathsf{GM}} := \big(sk^{(0)}_{\mathsf{AHO}}, sk^{(1)}_{\mathsf{AHO}}\big)$, $\mathcal{S}_{\mathsf{OA}} := \big(\beta_1, \beta_2\big)$ as authorities' private keys and define the group public key to be
$$\mathcal{Y} := \Big(g,\ pk^{(0)}_{\mathsf{AHO}},\ pk^{(1)}_{\mathsf{AHO}},\ mpk_{\mathsf{BB}},\ \mathbf{f},\ (U, V),\ \Sigma\Big).$$

Join$^{(\mathrm{GM}, \mathcal{U}_i)}$**:** the group manager and the prospective user $\mathcal{U}_i$ carry out the following interactive protocol $[\mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(\lambda, St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})]$:

1. $\mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y})$ chooses $x \xleftarrow{R} \mathbb{Z}_p$ and computes $X = g^x$ which is sent to $\mathsf{J}_{\mathsf{GM}}(\lambda, St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})$. If the value X already appears in some entry $\mathsf{transcript}_j$ of the database St_{trans}, J_{GM} aborts and returns $\perp$ to $\mathsf{J}_{\mathsf{user}}$.
2. J_{GM} assigns to $\mathcal{U}_i$ an available leaf v_i of the tree T and we let $\mathsf{ID}(v_i)$ be the identifier of v_i. Let $x_0 = \epsilon$, x_1, $\ldots$, $x_{\ell-1}$, $x_\ell = v_i$ be the path connecting the leaf v_i to the root ϵ of T. For $j = 0$ to ℓ, J_{GM} conducts the following steps.
   a. Compute an IBE private key $D_{x_j} = (D_{x_j,1}, D_{x_j,2}) = \Big(\big(h_0^{\mathsf{ID}(x_j)} \cdot h_1\big)^{r_{x_j}},\ g^{r_{x_j}}\Big)$ using a randomly chosen $r_{x_j} \xleftarrow{R} \mathbb{Z}_p$.
   b. Generate an AHO signature $\sigma_{x_j} = (\theta_{x_j,1}, \ldots, \theta_{x_j,7})$ on the pair $\big(X, D_{x_j,2}\big) \in \mathbb{G}^2$ so as to bind the node x_j and the value X that identifies $\mathcal{U}_i$.
3. J_{GM} sends the IBE private keys $\{D_{x_j}\}_{j=0}^{\ell}$ to $\mathsf{J}_{\mathsf{user}}$ that verifies their validity. If all keys are well-formed, $\mathsf{J}_{\mathsf{user}}$ acknowledges these values by generating a digital signature $sig_i = \mathrm{Sign}_{\mathsf{usk}[i]}\big(X \| \{D_{x_j}\}_{j=0}^{\ell}\big)$ and sends it back to J_{GM}.
4. J_{GM} checks that $\mathrm{Verify}_{\mathsf{upk}[i]}\big(X \| \{D_{x_j}\}_{j=0}^{\ell}, sig_i\big) = 1$. If not J_{GM} aborts. Otherwise, J_{GM} sends the AHO signatures $\{\sigma_{x_j}\}_{j=0}^{\ell}$ to $\mathsf{J}_{\mathsf{user}}$ and stores $\mathsf{transcript}_i = (X, \{D_{x_j}, \sigma_{x_j}\}_{j=0}^{\ell}, sig_i)$ in St_{trans}.
5. $\mathsf{J}_{\mathsf{user}}$ defines the membership certificate cert_i as $\mathsf{cert}_i = \big(\langle v_i \rangle, \{D_{x_j}, \sigma_{x_j}\}_{j=0}^{\ell}, X\big)$. The membership secret sec_i is defined to be $\mathsf{sec}_i = x$.

Revoke$(\mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, t, \mathcal{R}_t)$**:**

1. Parse $\mathcal{S}_{\mathsf{GM}}$ as $\mathcal{S}_{\mathsf{GM}} = (sk_{\mathsf{AHO}}^{(0)}, sk_{\mathsf{AHO}}^{(1)})$.
2. Using the CS covering algorithm, find a cover of the unrevoked user set $\{1, \ldots, N\} \backslash \mathcal{R}_t$ as the union of m sub-trees $S_1, \ldots, S_m$, with $m \leq r \cdot \log(N/r)$. Let $u_1, \ldots, u_m$ be the roots of these sub-trees
3. For $i = 1$ to m, do the following.
   a. Compute an IBE ciphertext $C_i = \big(h_0^{\mathsf{ID}(u_i)} \cdot h_1\big)$ for the identity $\mathsf{ID}(u_i)$.
   b. To authenticate C_i and bind it to the current revocation epoch t, use $sk_{\mathsf{AHO}}^{(1)}$ to generate an AHO signature $\Theta_i = (\Theta_{i,1}, \ldots, \Theta_{i,7}) \in \mathbb{G}^7$ on the pair $(C_i, g^t) \in \mathbb{G}^2$, where the epoch number t is interpreted as an element of $\mathbb{Z}_p$.

Return the revocation data RL_t which is defined to be

$$RL_t = \Big(t,\ \mathcal{R}_t,\ \{\mathsf{ID}(u_i), (C_i, \Theta_i)\}_{i=1}^{m}\Big) \tag{14}$$

Sign$(\mathcal{Y}, t, RK_t, \mathsf{cert}_i, \mathsf{sec}_i, M)$**:** return $\perp$ if $i \in \mathcal{R}_t$. Otherwise, to sign $M \in \{0,1\}^*$, generate a one-time key pair $(\mathsf{SK}, \mathsf{VK}) \leftarrow \mathcal{G}(\lambda)$. Parse cert_i and sec_i as $\big(\langle v_i \rangle, \{D_{x_j}, \sigma_{x_j}\}_{j=0}^{\ell}, X\big)$ and $x \in \mathbb{Z}_p$, respectively. Then, $\mathcal{U}_i$ conducts the following steps.

1. Using RL_t, determine the sub-tree S_l with $l \in \{1, \ldots, m\}$, that contains the leaf v_i (this subset must exist since $i \notin \mathcal{R}_t$) and let u_l be the root of S_l. Since u_l is an ancestor of v_i, the signer $\mathcal{U}_i$ necessarily knows and IBE private key of the form

$$D_{u_l} = (D_{u_l,1}, D_{u_l,2}) = \Big(\big(h_0^{\mathsf{ID}(u_l)} \cdot h_1\big)^{r_{u_l}},\ g^{r_{u_l}}\Big). \tag{15}$$

2. To prove that he holds a valid IBE private key for $C_l = \big(h_0^{\mathsf{ID}(u_l)} \cdot h_1\big)$, $\mathcal{U}_i$ first generates a commitment com_{C_l} to C_l. Then, he re-randomizes the corresponding signature $\Theta_l = (\Theta_{l,1}, \ldots, \Theta_{l,7})$ to

obtain $\{\Theta'_{l,i}\}_{i=1}^{7} \leftarrow \mathsf{ReRand}(pk_{\mathsf{AHO}}^{(1)}, \Theta_l)$ and computes commitments $\{com_{\Theta'_{l,i}}\}_{i\in\{1,2,5\}}$ to the resulting $\{\Theta'_{l,i}\}_{i\in\{1,2,5\}}$. Finally, he generates a proof π_{C_l} that C_l is a certified HIBE ciphertext for epoch t: *i.e.*, π_{C_l} provides evidence that

$$\begin{aligned} A^{(1)} \cdot e(\Theta'_{l,3}, \Theta'_{l,4})^{-1} \cdot e(G_2^{(1)}, g^t)^{-1} &= e(G_z^{(1)}, \Theta'_{l,1}) \cdot e(G_r^{(1)}, \Theta'_{l,2}) \cdot e(G_1^{(1)}, C_l), \\ B^{(1)} \cdot e(\Theta'_{l,6}, \Theta'_{l,7})^{-1} \cdot e(H_2^{(1)}, g^t)^{-1} &= e(H_z^{(1)}, \Theta'_{l,1}) \cdot e(H_r^{(1)}, \Theta'_{l,5}) \cdot e(H_1^{(1)}, C_l), \end{aligned} \tag{16}$$

Then, $\mathcal{U}_i$ generates commitments $com_{D_{u_l,1}}$ and $com_{D_{u_l,2}}$ to $D_{u_l,1}$ and $D_{u_l,2}$. Then, he generates a proof $\pi_{D_{u_l}}$ that $e(D_{u_l,1}, g) = e(C_l, D_{u_l,2})$. Since $\{\Theta'_{l,i}\}_{i\in\{3,4,6,7\}}$ are constants, the two relations of (16) are linear equations and π_{C_l} costs 6 elements while $\pi_{D_{u_l}}$ takes 9 elements.

3. Let $\sigma_{u_l} = (\theta_{u_l,1}, \ldots, \theta_{u_l,7}) \in \mathbb{G}^7$ be the structure-preserving signature on $(X, D_{u_l,2})$. Re-randomize σ_{u_l} to obtain $\{\theta'_{u_l,i}\}_{i=1}^{7} \leftarrow \mathsf{ReRand}(pk_{\mathsf{AHO}}^{(0)}, \sigma_{u_l})$. Then, generate commitments $\{com_{\theta'_{u_l,i}}\}_{i\in\{1,2,5\}}$ to $\{\theta'_{u_l,i}\}_{i\in\{1,2,5\}}$ as well as a commitment com_X to X. Finally, generate a proof $\pi_{\sigma_{u_l}}$ that committed variables $\{\theta'_{u_l,i}\}_{i\in\{1,2,5\}}$, X and $D_{u_l,2}$ satisfy the verification equations

$$\begin{aligned} A^{(0)} \cdot e(\theta'_{u_l,3}, \theta'_{u_l,4})^{-1} &= e(G_z^{(0)}, \theta'_{u_l,1}) \cdot e(G_r^{(0)}, \theta'_{u_l,2}) \cdot e(G_1^{(0)}, X) \cdot e(G_2^{(0)}, D_{u_l,2}), \\ B^{(0)} \cdot e(\theta'_{u_l,6}, \theta'_{u_l,7})^{-1} &= e(H_z^{(0)}, \theta'_{u_l,1}) \cdot e(H_r^{(0)}, \theta'_{u_l,5}) \cdot e(H_1^{(0)}, X) \cdot e(H_2^{(0)}, D_{u_l,2}). \end{aligned}$$

Since these equations are linear, $\pi_{\sigma_{u_l}}$ requires 6 group elements.

4. Compute a tag-based encryption of X by drawing $z_1, z_2 \xleftarrow{R} \mathbb{Z}_p$ and setting

$$(\Psi_1, \Psi_2, \Psi_3, \Psi_4, \Psi_5) = \big(f_1^{z_1}, f_2^{z_2}, X \cdot g^{z_1+z_2}, (g^{\mathsf{VK}} \cdot U)^{z_1}, (g^{\mathsf{VK}} \cdot V)^{z_2}\big).$$

5. Generate a NIZK proof that the commitment $com_X = (1, 1, X) \cdot \vec{f_1}^{\phi_{X,1}} \cdot \vec{f_2}^{\phi_{X,2}} \cdot \vec{f_3}^{\phi_{X,3}}$ and (Ψ_1, Ψ_2, Ψ_3) are BBS encryptions of the same value X. If we write $\vec{f_3} = (f_{3,1}, f_{3,2}, f_{3,3})$, com_X can be written as $(f_1^{\phi_{X,1}} \cdot f_{3,1}^{\phi_{X,3}}, f_2^{\phi_{X,2}} \cdot f_{3,2}^{\phi_{X,3}}, X \cdot g^{\phi_{X,1}+\phi_{X,2}} \cdot f_{3,3}^{\phi_{X,3}})$ and, given that $(\Psi_1, \Psi_2, \Psi_3) = (f_1^{z_1}, f_2^{z_2}, X \cdot g^{z_1+z_2})$, we have

$$com_X \odot (\Psi_1, \Psi_2, \Psi_3)^{-1} = \big(f_1^{\tau_1} \cdot f_{3,1}^{\tau_3},\ f_2^{\tau_2} \cdot f_{3,2}^{\tau_3},\ g^{\tau_1+\tau_2} \cdot f_{3,3}^{\tau_3}\big) \tag{17}$$

with $\tau_1 = \phi_{X,1} - z_1$, $\tau_2 = \phi_{X,2} - z_2$, $\tau_3 = \phi_{X,3}$. The signer $\mathcal{U}_i$ commits to the exponents $\{\tau_i\}_{i=1}^{3}$ (by computing $com_{\tau_j} = \vec{\varphi}^{\tau_j} \cdot \vec{f_1}^{\phi_{\tau_j,1}} \cdot \vec{f_2}^{\phi_{\tau_j,2}}$ for $j \in \{1, 2, 3\}$, using the vector $\vec{\varphi} = \vec{f_3} \cdot (1, 1, g)$), and generates proofs $\pi_{\textit{eq-com},1}$, $\pi_{\textit{eq-com},2}$ and $\pi_{\textit{eq-com},3}$ that $\{\tau_i\}_{i=1}^{3}$ satisfy the relations (17). Since (17) are linear equations, proofs $\{\pi_{\textit{eq-com},j}\}_{j=1}^{3}$ cost 2 elements each.

6. Compute $\sigma_{\mathsf{VK}} = g^{1/(x+\mathsf{VK})}$ and generate a commitment $com_{\sigma_{\mathsf{VK}}}$ to σ_{VK}. Then, generate a NIWI proof $\pi_{\sigma_{\mathsf{VK}}}$ that committed variables σ_{VK} and X satisfy

$$e(\sigma_{\mathsf{VK}}, X \cdot g^{\mathsf{VK}}) = e(g, g) \tag{18}$$

Relation (18) is a quadratic pairing product equation and requires a proof consisting of 9 group elements.

7. Using SK, generate a one-time signature $\sigma_{ots} = \mathcal{S}(\mathsf{SK}, (M, RL_t, \Psi_1, \Psi_2, \Psi_3, \Psi_4, \Psi_5, \Omega, \mathbf{com}, \mathbf{\Pi}))$ where $\Omega = \{\Theta'_{l,i}, \theta'_{u_l,i}\}_{i\in\{3,4,6,7\}}$ and

$$\begin{aligned} \mathbf{com} &= \big(com_{C_l}, \{com_{D_{u_l,j}}\}_{j=1}^{2}, com_X, \{com_{\Theta'_{l,i}}\}_{i\in\{1,2,5\}}, \{com_{\theta'_{u_l,j}}\}_{j\in\{1,2,5\}}, \{com_{\tau_j}\}_{j=1}^{3}, com_{\sigma_{\mathsf{VK}}}\big) \\ \mathbf{\Pi} &= (\pi_{C_l}, \pi_{D_{u_l}}, \pi_{\sigma_{u_l}}, \pi_{\textit{eq-com},1}, \pi_{\textit{eq-com},2}, \pi_{\textit{eq-com},3}, \pi_{\sigma_{\mathsf{VK}}}). \end{aligned}$$

Return the signature

$$\sigma = \big(\mathsf{VK}, \Psi_1, \Psi_2, \Psi_3, \Psi_4, \Psi_5, \Omega, \mathbf{com}, \mathbf{\Pi}, \sigma_{ots}\big). \tag{19}$$

Verify$(\sigma, M, t, RL_t, \mathcal{Y})$**:** parse σ as above and return 1 if and only if the following checks all succeed.

1. If $\mathcal{V}(\mathsf{VK}, (M, RL_t, \Psi_1, \Psi_2, \Psi_3, \Psi_4, \Psi_5, \Omega, \mathbf{com}, \mathbf{\Pi}), \sigma_{ots}) = 0$, return 0.
2. Return 0 if $e(\Psi_1, g^{\mathsf{VK}} \cdot U) \neq e(f_1, \Psi_4)$ or $e(\Psi_2, g^{\mathsf{VK}} \cdot V) \neq e(f_2, \Psi_5)$.
3. Return 1 if all proofs properly verify. Otherwise, return 0.

Open$(M, t, RL_t, \sigma, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, St)$**:** given $\mathcal{S}_{\mathsf{OA}} = (\beta_1, \beta_2)$, parse the signature σ as in (19) and return $\perp$ if $\mathsf{Verify}(\sigma, M, t, RL_t, \mathcal{Y}) = 0$. Otherwise, compute $\tilde{X} = \Psi_3 \cdot \Psi_1^{-1/\beta_1} \cdot \Psi_2^{-1/\beta_2}$. Find a record

$$\langle i, \mathsf{transcript}_i = (X, \{D_{x_j}, \sigma_{x_j}\}_{j=0}^{\ell}, sig_i)\rangle$$

such that $X = \tilde{X}$. If no such record exists in St_{trans}, return $\perp$. Otherwise, return i.

The size of signatures is exactly the same as in the construction based on the SD method. Revocation lists have become longer: they now contain $O(r \cdot \log(N/r))$ group elements (as in Section 3, the representation of $\mathsf{ID}(u_i)$ is at least as short as that of a group element in RL_t) and they can be seen as ciphertexts in the public-key variant [34] of the CS method. On the other hand, we note that membership certificates now consist of $O(\log N)$ group elements (vs $O(\log^3 N)$ in the SD method).

The complexity of the verification algorithm does not depend on r or N. As for the signing algorithm, it first requires $O(\log \log N)$ combinatorial operations (see [51]) to determine which sub-tree the signer is a leaf of. However, the cost of these operations (which are only needed once per epoch) is small compared to that of public-key arithmetic operations. As we can see, the number of arithmetic operations is independent of r and N when it comes to generate or verify signatures.

E.1 Security

All security proofs go through essentially without changes. The proof of Theorem 4 relies on the key-robustness of the Boneh-Boyen IBE [14], but this property is implied by Lemma 1: indeed, the first IBE scheme of [13] (in its single-level variant) can be seen as a single-level variant of the Boneh-Boyen-Goh HIBE.

Theorem 4 (Misidentification). *The scheme is secure against misidentification attacks assuming that the q-SFP problem is hard for* $q = \max(\ell \cdot q_a, q_r^2)$, *where* q_a *and* q_r *denote the maximal numbers of* $Q_{\mathsf{a\text{-}join}}$ *queries and* Q_{revoke} *queries, respectively, and* $\ell = \log N$.

Proof. The proof is almost identical to that of Theorem 1. The only difference is that, in the treatment of Type II.a forgeries, the simulator $\mathcal{B}$ has to generate at most $\ell \cdot q_a$ AHO signatures overall (rather than $\ell^2 \cdot q_a$ in the proof of Theorem 1). □

Theorem 5 (Non-frameability). *The scheme is secure against framing attacks assuming that: (i) the* q_b*-SDH assumption holds in* $\mathbb{G}$, *where* q_b *is the maximal number of* $Q_{\mathsf{b\text{-}join}}$*-queries; (ii)* Σ *is a strongly unforgeable one-time signature.*

Proof. The proof is the same as the proof of Theorem 2. □

Theorem 6 (Anonymity). *The advantage of any anonymity adversary is at most*

$$\mathbf{Adv}^{\mathrm{anon}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{ots}}(\lambda) + 3 \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\lambda).$$

Proof. The proof is completely identical to the proof of Theorem 3. □

Group Signatures with Almost-for-free Revocation

Benoît Libert[1] *, Thomas Peters[1] **, and Moti Yung[2]

[1]Université catholique de Louvain, ICTEAM Institute (Belgium)
[2] Google Inc. and Columbia University (USA)

Abstract. Group signatures are a central cryptographic primitive where users can anonymously and accountably sign messages in the name of a group they belong to. Several efficient constructions with security proofs in the standard model (*i.e.*, without the random oracle idealization) appeared in the recent years. However, like standard PKIs, group signatures need an efficient revocation system to be practical. Despite years of research, membership revocation remains a non-trivial problem: many existing solutions do not scale well due to either high overhead or constraining operational requirements (like the need for all users to update their keys after each revocation). Only recently, Libert, Peters and Yung (Eurocrypt'12) suggested a new scalable revocation method, based on the Naor-Naor-Lotspiech (NNL) broadcast encryption framework, that interacts nicely with techniques for building group signatures in the standard model. While promising, their mechanism introduces important storage requirements at group members. Namely, membership certificates, which used to have constant size in existing standard model constructions, now have polylog size in the maximal cardinality of the group (NNL, after all, is a tree-based technique and such dependency is naturally expected). In this paper we show how to obtain private keys of *constant* size. To this end, we introduce a new technique to leverage the NNL subset cover framework in the context of group signatures but, perhaps surprisingly, without logarithmic relationship between the size of private keys and the group cardinality. Namely, we provide a way for users to efficiently prove their membership of one of the generic subsets in the NNL subset cover framework. This technique makes our revocable group signatures competitive with ordinary group signatures (*i.e.*, without revocation) in the standard model. Moreover, unrevoked members (as in PKIs) still do not need to update their keys at each revocation.

Keywords. Group signatures, revocation, standard model, efficiency, short private keys.

1 Introduction

Group signatures, as suggested by Chaum and van Heyst [29], allow members of a group managed by some authority to sign messages in the name of the group while hiding their identity. At the same time, a tracing authority has the power of identifying the signer if necessary. A crucial problem is the revocation of the anonymous signing capability of users when they are banned from or intentionally leave the group.

1.1 Related Work

Ordinary group signatures. The first efficient and provably coalition-resistant group signature dates back to the work of Ateniese, Camenisch, Joye and Tsudik [6]. By the time their scheme appeared, the security of the primitive was not appropriately formalized yet. Suitable security definitions remained lacking until the work of Bellare, Micciancio and Warinschi [8] (BMW) who captured all the requirements of group signatures in three properties. In (a variant of) this model,

* This author acknowledges the Belgian Fund for Scientific Research (F.R.S.-F.N.R.S.) for his "Collaborateur scientifique" fellowship.

** Supported by the IUAP B-Crypt Project and the Walloon Region Camus Project.

Boneh, Boyen and Shacham [14] obtained very short signatures using the random oracle methodology [9].

The BMW model assumes static groups where no new member can be introduced after the setup phase. The setting of dynamically changing groups was analyzed later on by Bellare-Shi-Zhang [10] and, independently, by Kiayias and Yung [40]. In the models of [10, 40], constructions featuring relatively short signatures were proposed in [54, 30]. A construction in the standard model was also suggested by Ateniese *et al.* [5] under interactive assumptions. At the same time, Boyen and Waters gave a different solution [18] without random oracles using more standard assumptions. By improving upon their own scheme, they managed [19] to obtain signatures of constant size. Their constructions [18, 19] were both presented in the BMW model [8] and provide anonymity in the absence of signature opening oracle. In the dynamic model [10], Groth [34] showed a system in the standard model with $O(1)$-size signatures but, due to very large hidden constants, his scheme was mostly a feasibility result. Later on, Groth came up with an efficient realization [35] (and signatures of about 50 group elements) with the strongest anonymity level.

Revocation. As in ordinary PKIs, where certificate revocation is a critical issue, membership revocation is a complex problem that has been extensively studied [20, 7, 26, 17] in the last decade. Generating a new group public key and distributing new signing keys to unrevoked members is a simple solution. In large groups, it is impractical to update the public key and provide members with new keys after they joined the group. Bresson and Stern suggested a different approach [20] consisting of having the signer prove that his membership certificate does not belong to a list of revoked certificates. Unfortunately, the length of signatures grows with the number of revoked members. In forward-secure group signatures, Song [56] chose a different way to handle revocation but verification takes linear time in the number of excluded users.

Camenisch and Lysyanskaya [26] proposed an elegant method using accumulators[1] [11]. Their technique, also used in [59, 24], allows revoking members while keeping $O(1)$ costs for signing and verifying. The downside of this approach is its history-dependence: it requires users to follow the dynamic evolution of the group and keep track of all changes: each revocation incurs a modification of the accumulator value, so that unrevoked users have to upgrade their membership certificate before signing new messages. In the worst case, this may require up to $O(r)$ exponentiations, if r is the number of revoked users.

Another drawback of accumulator-based approaches is their limited applicability in the standard model. Indeed, for compatibility reasons with the central tool of Groth-Sahai proofs, pairing-based accumulators are the only suitable candidates. However, in known pairing-based accumulators [53, 24], public keys have linear size in the maximal number of accumulations, which would result in linear-size group public keys in immediate implementations. To address this concern in delegatable anonymous credentials, Acar and Nguyen [4] chose to sacrifice the constant size of proofs of non-membership but, in group signatures, this would prevent signatures from having constant size. Boneh, Boyen and Shacham [14] managed to avoid linear dependencies in a revocation mechanism along the lines of [26]. Unfortunately, their technique does not seem to readily interact[2] with Groth-

[1] An accumulator is a kind of "hash" function mapping a set of values to a short, constant-size string while allowing to efficiently prove that a specific value was accumulated.

[2] In [14], signing keys consist of pairs $(g^{1/(\omega+s)}, s) \in \mathbb{G} \times \mathbb{Z}_p$, where $\omega \in \mathbb{Z}_p$ is the secret key of the group manager, and the revocation method relies on the availability of the exponent $s \in \mathbb{Z}_p$. In the standard model, the Groth-Sahai techniques would require to turn the membership certificates into triples $(g^{1/(\omega+s)}, g^s, u^s)$, for some $u \in \mathbb{G}$ (as in [19]), which is not compatible with the revocation mechanism.

Sahai proofs [36] so as to work in the standard model.

In [21], Brickell considered the notion of *verifier-local revocation* group signatures, for which formal definitions were given by Boneh and Shacham [17] and other extensions were proposed in [50, 61, 45]. In this approach, revocation messages are only sent to verifiers and the signing algorithm is completely independent of the number of revocations. Verifiers take as additional input a revocation list (RL), maintained by the group manager, and have to perform a revocation test for each RL entry in order to be convinced that signatures were not issued by a revoked member (a similar revocation mechanism is used in [22]). The verification cost is thus inevitably linear in the number of expelled users.

In 2009, Nakanishi, Fuji, Hira and Funabiki [49] came up with a revocable group signature with constant complexities for signing/verifying. At the same time, group members never have to update their keys. On the other hand, their proposal suffers from linear-size group public keys in the maximal number N of users, although a variant reduces the group public key size to $O(N^{1/2})$.

In anonymous credentials, Tsang *et al.* [57, 58] showed how to prevent users from anonymously authenticating themselves without compromising their anonymity or involving a trusted third party. Their schemes either rely on accumulators (which may be problematic in our setting) or have linear proving complexity in the number of revocations. Camenisch, Kohlweiss and Soriente [25] dealt with revocations in anonymous credentials by periodically updating users credentials in which a specific attribute indicates a validity period. In group signatures, their technique would place an important burden on the group manager who would have to generate updates for each unrevoked individual credential.

While, for various reasons, none of the above constructions conveniently supports large groups, a highly scalable revocation mechanism borrowed from the literature on broadcast encryption was recently described by Libert, Peters and Yung [47] (LPY). Using the Subset Cover framework of Naor, Naor and Lotspiech [51] (NNL), they described a history-independent revocable group signature in the standard model with constant verification time and at most polylogarithmic complexity in other parameters. The technique of [47] blends well with structure-preserving signatures [1, 2] and the Groth-Sahai proofs [36]. The best tradeoff of [47] builds on the Subset Difference (SD) method [51] in its public-key variant due to Dodis and Fazio [31]. It features constant signature size and verification time, $O(\log N)$-size group public keys, revocation lists of size $O(r)$ (as in standard PKIs and group signatures with verifier-local revocation) and membership certificates of size $O(\log^3 N)$. This can be reduced to $O(\log N)$ using the Complete Subtree method [51] but revocation lists are then inflated by a factor of $O(\log N/r)$. Although the Layered Subset Difference method [37] allows for noticeable improvements, the constructions of [47] suffer from relatively large membership certificates. However, some logarithmic dependency on the group size is expected when basing revocation on a tree-like NNL methodology.

1.2 Our Contributions

As mentioned above, to date, in the only scalable revocable group signatures with constant verification time in the standard model [47], group members have to store a polylogarithmic number of group elements. In many applications, however, this can rapidly become unwieldy even for moderately large groups: for example, using the Subset Difference method with $N = 1000 \approx 2^{10}$, users may have to privately store thousands of group elements. In order to be competitive with other group signatures in the standard model such as [35] and still be able to revoke members while keeping them “stateless”, it is highly desirable to reduce this complexity.

In this paper, we start with the approach of [47] so as to instantiate the Subset Difference method, but obtain private keys of *constant* size without degrading other performance criteria. This may sound somewhat surprising since, in the SD method, (poly)logarithmic complexities inherently seem inevitable in several metrics. Indeed, in the context of broadcast encryption [51], it requires private keys of size $O(\log^2 N)$ (and even $O(\log^3 N)$ in the public key setting [31] if the result of Boneh-Boyen-Goh [13] is used). Here, we reduce this overhead to a constant while the only dependency on N is a $O(\log N)$-size group public key.

The key idea is as follows. As in the NNL framework, group members are assigned to a leaf of a binary tree and each unrevoked member should belong to exactly one subset in the cover of authorized leafs determined by the group manager. Instead of relying on hierarchical identity-based encryption [15, 38, 33] as in the public-key variant [31] of NNL, we use a novel way for users to non-interactively prove their membership of some generic subset of the SD method using a proof of constant size.

To construct these "compact anonymous membership proofs", we employ *concise* vector commitment schemes [46, 27], where each commitment can be opened w.r.t. individual coordinates in a space-efficient manner (namely, the size of a coordinate-wise opening does not depend on the length of the vector). These vector commitments interact nicely with the specific shape of subsets – as differences between two subtrees – in the SD method. Using them, we compactly encode as a vector the path from the user's leaf to the root. To provide evidence of their inclusion in one of the SD subsets, group members successively prove the equality and the inequality between two coordinates of their vector (*i.e.*, two nodes of the path from their leaf to the root) and specific node labels indicated by an appropriate entry of the revocation list. This is where the position-wise openability of concise commitments is very handy. Of course, for anonymity purposes, the relevant entry of the revocation list only appears in committed form in the group signature. In order to prove that he is using a legal entry of the revocation list, the user generates a set membership proof [23] and proves knowledge of a signature from the group manager on the committed RL entry.

Our technique allows making the most of the LPY approach [47] by reducing the size of membership certificates to a small constant: at the cost of lengthening signatures by a factor of only 1.5, we obtain membership certificates consisting of only 9 group elements and a small integer. For $N = 1000$, users' private keys are thus compressed by a multiplicative factor of several hundreds and this can only become more dramatic for larger groups. At the same time, our main scheme retains all the useful properties of [47]: like the construction of Nakanishi *et al.* [49], it does not require users to update their membership certificates at any time but, unlike [49], our group public key size is $O(\log N)$. Like the SD-based construction of [47], our system uses revocation lists of size $O(r)$, which is on par with Certificate Revocation Lists (CRLs) of standard PKIs. It is worth noting that RLs are *not* part of the group public key: verifiers only need to know the number of the latest revocation epoch and they should not bother to read RLs entirely.

Eventually, our novel approach yields revocable group signatures that become competitive with the regular CRL approach in PKIs: signature generation and verification have constant cost, signatures and membership certificates being of $O(1)$-size while revocation lists have size $O(r)$. A detailed efficiency comparison with previous approaches is given in Section 4. Finally, it is conceivable that our improved revocation technique can find applications beyond group signatures.

2 Background

2.1 Bilinear Maps and Complexity Assumptions

We use bilinear maps $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ over groups of prime order p where $e(g, h) \neq 1_{\mathbb{G}_T}$ if and only if $g, h \neq 1_{\mathbb{G}}$. In these groups, we rely on hardness assumptions that are all falsifiable [52].

Definition 1 ([14]). *The* **Decision Linear Problem** *(DLIN) in $\mathbb{G}$, is to distinguish the distributions $(g^a, g^b, g^{ac}, g^{bd}, g^{c+d})$ and $(g^a, g^b, g^{ac}, g^{bd}, g^z)$, with $a, b, c, d \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$, $z \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$. The* **Decision Linear Assumption** *is the intractability of DLIN for any PPT distinguisher* D.

Definition 2 ([12]). *The* **q-Strong Diffie-Hellman problem** *(q-SDH) in $\mathbb{G}$ is, given a tuple $(g, g^a, \ldots, g^{(a^q)})$, for some $g \stackrel{R}{\leftarrow} \mathbb{G}$ and $a \stackrel{R}{\leftarrow} \mathbb{Z}_p$, to find a pair $(g^{1/(a+s)}, s) \in \mathbb{G} \times \mathbb{Z}_p$.*

We use a signature scheme proposed by Abe *et al.* [1], the security of which relies on this assumption.

Definition 3 ([1]). *In a group $\mathbb{G}$, the* **q-Simultaneous Flexible Pairing Problem** *(q-SFP) is, given $\big(g_z,\ h_z,\ g_r,\ h_r,\ a,\ \tilde{a},\ b,\ \tilde{b} \in \mathbb{G}\big)$ and $q \in \mathsf{poly}(\lambda)$ tuples $(z_j, r_j, s_j, t_j, u_j, v_j, w_j) \in \mathbb{G}^7$ such that*

$$e(a, \tilde{a}) = e(g_z, z_j) \cdot e(g_r, r_j) \cdot e(s_j, t_j) \quad \text{and} \quad e(b, \tilde{b}) = e(h_z, z_j) \cdot e(h_r, u_j) \cdot e(v_j, w_j), \tag{1}$$

to find a tuple $(z^\star, r^\star, s^\star, t^\star, u^\star, v^\star, w^\star) \in \mathbb{G}^7$ satisfying relation (1) and such that $z^\star \notin \{1_{\mathbb{G}}, z_1, \ldots, z_q\}$.

The paper will appeal to two other assumptions. The first one was implicitly introduced in [16].

Definition 4 ([16]). *Let $\mathbb{G}$ be a group of prime order p. The* **ℓ-Diffie-Hellman Exponent** *(ℓ-DHE) problem is, given elements $(g, g_1, \ldots, g_\ell, g_{\ell+2}, \ldots, g_{2\ell}) \in \mathbb{G}^{2\ell}$ such that $g_i = g^{(\alpha^i)}$ for each i and where $\alpha \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$, to compute the missing element $g_{\ell+1} = g^{(\alpha^{\ell+1})}$.*

We actually need a stronger variant, used in [39], of the ℓ-DHE assumption. The Flexible Diffie-Hellman assumption [43] asserts the hardness of finding a non-trivial triple $(g^\mu, g^{a\cdot\mu}, g^{ab\cdot\mu})$, for some non-zero $\mu \in \mathbb{Z}_p^*$, given (g, g^a, g^b). The following assumption relaxes the ℓ-DHE assumption in a similar way.

Definition 5. *In a group $\mathbb{G}$ of prime order p, the* **Flexible ℓ-Diffie-Hellman Exponent** *(ℓ-FlexDHE) problem is, given $(g, g_1, \ldots, g_\ell, g_{\ell+2}, \ldots, g_{2\ell}) \in \mathbb{G}^{2\ell}$ such that $g_i = g^{(\alpha^i)}$ for each i and where $\alpha \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$, to compute a non-trivial triple $(g^\mu, g_{\ell+1}^\mu, g_{2\ell}^\mu) \in (\mathbb{G}\backslash\{1_{\mathbb{G}}\})^3$, for some $\mu \in \mathbb{Z}_p^*$ and where $g_{\ell+1} = g^{(\alpha^{\ell+1})}$.*

The reason why we need to rely on the above assumption instead of the weaker ℓ-DHE assumption is that, in our proofs, the exponent $\mu \in \mathbb{Z}_p$ will appear inside Groth-Sahai commitments [36], from which only values of the form $(g^\mu, g_{\ell+1}^\mu)$ will be efficiently extractable. The additional element $g_{2\ell}^\mu$ will thus prevent the adversary from simply choosing $\mu = \alpha$ or $\mu = \alpha^{-1}$.

A proof of the generic hardness of the ℓ-FlexDHE problem is given in [39]. We note that, while the strength of the assumption grows with ℓ, ℓ is only logarithmic in the maximal number of users here.

2.2 Groth-Sahai Proof Systems

The fundamental Groth-Sahai (GS) techniques [36] can be based on the DLIN assumption, where they use prime order groups and a common reference string containing three vectors $\vec{f_1}, \vec{f_2}, \vec{f_3} \in \mathbb{G}^3$, where $\vec{f_1} = (f_1, 1, g)$, $\vec{f_2} = (1, f_2, g)$ for some $f_1, f_2 \in \mathbb{G}$. To commit to a group element $X \in \mathbb{G}$, one chooses $r, s, t \xleftarrow{R} \mathbb{Z}_p^*$ and computes $\vec{C} = (1, 1, X) \cdot \vec{f_1}^{\,r} \cdot \vec{f_2}^{\,s} \cdot \vec{f_3}^{\,t}$. In the perfect soundness setting, we have $\vec{f_3} = \vec{f_1}^{\,\xi_1} \cdot \vec{f_2}^{\,\xi_2}$ where $\xi_1, \xi_2 \in \mathbb{Z}_p^*$. Commitments $\vec{C} = (f_1^{r+\xi_1 t}, f_2^{s+\xi_2 t}, X \cdot g^{r+s+t(\xi_1+\xi_2)})$ are then extractable (and distributed as Boneh-Boyen-Shacham (BBS) ciphertexts [14]) using $\beta_1 = \log_g(f_1)$, $\beta_2 = \log_g(f_2)$. In the witness indistinguishability (WI) setting, vectors $\vec{f_1}, \vec{f_2}, \vec{f_3}$ are linearly independent and $\vec{C}$ is a perfectly hiding commitment. Under the DLIN assumption, the two kinds of CRS are computationally indistinguishable.

To commit to an exponent $x \in \mathbb{Z}_p$, one computes $\vec{C} = \vec{\varphi}^{\,x} \cdot \vec{f_1}^{\,r} \cdot \vec{f_2}^{\,s}$, where $r, s \xleftarrow{R} \mathbb{Z}_p^*$, using a CRS consisting of vectors $\vec{\varphi}, \vec{f_1}, \vec{f_2}$. In the perfect soundness setting, $\vec{\varphi}, \vec{f_1}, \vec{f_2}$ are linearly independent ($\vec{\varphi}$ is often chosen as $\vec{\varphi} = \vec{f_3} \cdot (1, 1, g)$, where $\vec{f_3} = \vec{f_1}^{\,\xi_1} \cdot \vec{f_2}^{\,\xi_2}$, for example) whereas, in the WI setting, choosing $\vec{\varphi} = \vec{f_1}^{\,\xi_1} \cdot \vec{f_2}^{\,\xi_2}$ gives a perfectly hiding commitment since $\vec{C}$ is always a BBS encryption of $1_{\mathbb{G}}$.

To prove that committed variables satisfy a set of relations, the prover computes one commitment per variable and one proof element per relation. Such non-interactive witness indistinguishable (NIWI) proofs are available for pairing-product equations, which are relations of the type

$$\prod_{i=1}^{n} e(\mathcal{A}_i, \mathcal{X}_i) \cdot \prod_{i=1}^{n} \cdot \prod_{j=1}^{n} e(\mathcal{X}_i, \mathcal{X}_j)^{a_{ij}} = t_T, \tag{2}$$

for variables $\mathcal{X}_1, \ldots, \mathcal{X}_n \in \mathbb{G}$ and constants $t_T \in \mathbb{G}_T$, $\mathcal{A}_1, \ldots, \mathcal{A}_n \in \mathbb{G}$, $a_{ij} \in \mathbb{Z}_p$, for $i, j \in \{1, \ldots, n\}$. Efficient NIWI proofs also exist for multi-exponentiation equations, which are of the form

$$\prod_{i=1}^{m} \mathcal{A}_i^{y_i} \cdot \prod_{j=1}^{n} \mathcal{X}_j^{b_j} \cdot \prod_{i=1}^{m} \cdot \prod_{j=1}^{n} \mathcal{X}_j^{y_i \gamma_{ij}} = T, \tag{3}$$

for variables $\mathcal{X}_1, \ldots, \mathcal{X}_n \in \mathbb{G}$, $y_1, \ldots, y_m \in \mathbb{Z}_p$ and constants $T, \mathcal{A}_1, \ldots, \mathcal{A}_m \in \mathbb{G}$, $b_1, \ldots, b_n \in \mathbb{Z}_p$ and $\gamma_{ij} \in \mathbb{G}$, for $i \in \{1, \ldots, m\}, j \in \{1, \ldots, n\}$.

In pairing-product equations, proofs for quadratic equations consist of 9 group elements whereas linear equations (*i.e.*, where $a_{ij} = 0$ for all i, j in equation (2)) only demand 3 group elements each. Linear multi-exponentiation equations of the type $\prod_{i=1}^{m} \mathcal{A}_i^{y_i} = T$ demand 2 group elements.

Multi-exponentiation equations admit zero-knowledge (NIZK) proofs at no additional cost. On a simulated CRS (prepared for the WI setting), a trapdoor allows simulating proofs without using the witnesses and simulated proofs are distributed as real proofs.

2.3 Structure-Preserving Signatures

Many anonymity-related protocols (e.g., [28, 1, 2, 32, 3]) require to sign elements of bilinear groups while maintaining the feasibility of conveniently proving that a committed signature is valid for a committed message.

Abe, Haralambiev and Ohkubo [1, 2] (AHO) showed how to sign messages of n group elements using signatures consisting of $O(1)$ group elements. In the context of symmetric pairings, the description hereafter assumes public parameters $\mathsf{pp} = \big((\mathbb{G}, \mathbb{G}_T),\ g\big)$ consisting of groups $(\mathbb{G}, \mathbb{G}_T)$ of order $p > 2^\lambda$, where $\lambda \in \mathbb{N}$ is a security parameter, with a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ and a generator $g \in \mathbb{G}$.

Keygen(pp, n)**:** given an upper bound $n \in \mathbb{N}$ on the number of group elements per signed message, choose generators $G_r, H_r \overset{R}{\leftarrow} \mathbb{G}$. Pick $\gamma_z, \delta_z \overset{R}{\leftarrow} \mathbb{Z}_p$ and $\gamma_i, \delta_i \overset{R}{\leftarrow} \mathbb{Z}_p$, for $i = 1$ to n. Then, compute $G_z = G_r^{\gamma_z}$, $H_z = H_r^{\delta_z}$ and $G_i = G_r^{\gamma_i}$, $H_i = H_r^{\delta_i}$ for each $i \in \{1, \dots, n\}$. Finally, choose $\alpha_a, \alpha_b \overset{R}{\leftarrow} \mathbb{Z}_p$ and define $A = e(G_r, g^{\alpha_a})$ and $B = e(H_r, g^{\alpha_b})$. The public key is defined to be

$$pk = \big(G_r,\ H_r,\ G_z,\ H_z,\ \{G_i, H_i\}_{i=1}^n,\ A,\ B\big) \in \mathbb{G}^{2n+4} \times \mathbb{G}_T^2$$

while the private key is $sk = \big(\alpha_a, \alpha_b, \gamma_z, \delta_z, \{\gamma_i, \delta_i\}_{i=1}^n\big)$.

Sign($sk, (M_1, \dots, M_n)$)**:** to sign a vector $(M_1, \dots, M_n) \in \mathbb{G}^n$ using $sk = (\alpha_a, \alpha_b, \gamma_z, \delta_z, \{\gamma_i, \delta_i\}_{i=1}^n)$, choose $\zeta, \rho_a, \rho_b, \omega_a, \omega_b \overset{R}{\leftarrow} \mathbb{Z}_p$ and compute $\theta_1 = g^\zeta$ as well as

$$\theta_2 = g^{\rho_a - \gamma_z \zeta} \cdot \prod_{i=1}^n M_i^{-\gamma_i}, \qquad \theta_3 = G_r^{\omega_a}, \qquad \theta_4 = g^{(\alpha_a - \rho_a)/\omega_a},$$
$$\theta_5 = g^{\rho_b - \delta_z \zeta} \cdot \prod_{i=1}^n M_i^{-\delta_i}, \qquad \theta_6 = H_r^{\omega_b}, \qquad \theta_7 = g^{(\alpha_b - \rho_b)/\omega_b},$$

The signature consists of $\sigma = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7)$.

Verify($pk, \sigma, (M_1, \dots, M_n)$)**:** parse σ as $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7) \in \mathbb{G}^7$ and return 1 iff these equalities hold:

$$A = e(G_z, \theta_1) \cdot e(G_r, \theta_2) \cdot e(\theta_3, \theta_4) \cdot \prod_{i=1}^n e(G_i, M_i),$$
$$B = e(H_z, \theta_1) \cdot e(H_r, \theta_5) \cdot e(\theta_6, \theta_7) \cdot \prod_{i=1}^n e(H_i, M_i).$$

The scheme was proved [1, 2] existentially unforgeable under chosen-message attacks under the q-SFP assumption, where q is the number of signing queries.

Signatures can be publicly randomized to obtain a different signature $\{\theta_i'\}_{i=1}^7 \leftarrow \mathsf{ReRand}(pk, \sigma)$ on $(M_1, \dots, M_n)$. After randomization, we have $\theta_1' = \theta_1$ whereas other signature components $\{\theta_i'\}_{i=2}^7$ are uniformly distributed among the values satisfying $e(G_r, \theta_2') \cdot e(\theta_3', \theta_4') = e(G_r, \theta_2) \cdot e(\theta_3, \theta_4)$ and $e(H_r, \theta_5') \cdot e(\theta_6', \theta_7') = e(H_r, \theta_5) \cdot e(\theta_6, \theta_7)$. Moreover, $\{\theta_i'\}_{i \in \{3,4,6,7\}}$ are statistically independent of the message and the rest of the signature. This implies that, in privacy-preserving protocols, re-randomized $\{\theta_i'\}_{i \in \{3,4,6,7\}}$ can be safely given in the clear as long as $(M_1, \dots, M_n)$ and $\{\theta_i'\}_{i \in \{1,2,5\}}$ are given in committed form.

In [3], Abe, Groth, Haralambiev and Ohkubo described a more efficient structure-preserving signature based on interactive assumptions. Here, we only rest on non-interactive assumptions.

2.4 Vector Commitment Schemes

We use concise vector commitment schemes, where commitments can be opened with a short decommitment string for each individual coordinate. Such commitments based on ideas from [16, 24] were described by Libert and Yung [46] and, under weaker assumptions, by Catalano and Fiore [27]. In [46], the commitment key is $ck = (g, g_1, \ldots, g_\ell, g_{\ell+2}, \ldots, g_{2\ell}) \in \mathbb{G}^{2\ell}$, where $g_i = g^{(\alpha^i)}$ for each i. The trapdoor of the commitment is $g_{\ell+1}$, which does not appear in ck. To commit to a vector $\vec{m} = (m_1, \ldots, m_\ell)$, the committer picks $r \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and computes $C = g^r \cdot \prod_{\kappa=1}^{\ell} g_{\ell+1-\kappa}^{m_\kappa}$. A single group element $W_i = g_i^r \cdot \prod_{\kappa=1,\kappa\neq i}^{\ell} g_{\ell+1-\kappa+i}^{m_\kappa}$ provides evidence that m_i is the i-th component of $\vec{m}$ as it satisfies the relation $e(g_i, C) = e(g, W_i) \cdot e(g_1, g_\ell)^{m_i}$. The infeasibility of opening a commitment to two distinct messages for some coordinate i relies on the ℓ-DHE assumption. For our purposes, we only rely on the position-wise binding property of vector commitments and do not need them to be hiding. The randomizer r will thus be removed from the expression of C.

2.5 The NNL Framework for Broadcast Encryption

The important Subset Cover framework [51] considers secret-key broadcast encryption schemes with $N = 2^\ell$ registered receivers. Each receiver is associated with a leaf of a complete binary tree T of height ℓ where each node is assigned a secret key. If $\mathcal{N}$ denotes the universe of users and $\mathcal{R} \subset \mathcal{N}$ is the set of revoked receivers, the framework's idea is to partition the set of non-revoked users into m disjoint subsets $S_1, \ldots, S_m$ such that $\mathcal{N} \backslash \mathcal{R} = S_1 \cup \ldots \cup S_m$. Depending on the way to divide $\mathcal{N} \backslash \mathcal{R}$, different tradeoffs are possible.

The Subset Difference (SD) method yields a transmission cost of $O(|\mathcal{R}|)$ and a storage complexity in $O(\log^2 N)$. For each node $x_j \in \mathsf{T}$, we call T_{x_j} the subtree rooted at x_j. The unrevoked set $\mathcal{N} \backslash \mathcal{R}$ is partitioned into disjoint subsets $S_{k_1,u_1}, \ldots, S_{k_m,u_m}$. For each $i \in \{1, \ldots, m\}$, the subset S_{k_i,u_i} is determined by a node x_{k_i} and one of its descendants x_{u_i} – which are called *primary* and *secondary* roots of S_{k_i,u_i}, respectively – and it consists of the leaves of $\mathsf{T}_{x_{k_i}}$ that are not in $\mathsf{T}_{x_{u_i}}$. Each user belongs to many generic subsets, so that the number of subsets bounded by $m = 2 \cdot |\mathcal{R}| - 1$, as proved in [51].

In the broadcast encryption scenario, a sophisticated key distribution process is necessary to avoid a prohibitive storage overhead. Each subset S_{k_i,u_i} is assigned a "proto-key" $P_{x_{k_i},x_{u_i}}$ that allows deriving the actual symmetric encryption key K_{k_i,u_i} for S_{k_i,u_i} and as well as proto-keys $P_{x_{k_i},x_{u_l}}$ for any descendant x_{u_l} of x_{u_i}. Eventually, each user has to store $O(\log^2 N)$ keys. In the setting of group signatures, we will show that, somewhat unexpectedly, the use of vector commitment schemes allows reducing the private storage to a constant: the size of users' private keys only depends on the security parameter λ, and not on N.

2.6 Revocable Group Signatures

As in [49, 47] (and w.l.o.g.), we consider schemes that have their lifetime divided into revocation epochs at the beginning of which group managers update their revocation lists.

The syntax and the security model are similar to those used by Kiayias and Yung [40]. Like the Bellare-Shi-Zhang model [10], the Kiayias-Yung model assumes an interactive join protocol whereby the user becomes a group member by interacting with the group manager.

Syntax. We denote by $N \in \mathsf{poly}(\lambda)$ the maximal number of group members. At the beginning of each revocation epoch t, the group manager publicizes an up-to-date revocation list RL_t and we denote by $\mathcal{R}_t \subset \{1, \ldots, N\}$ the corresponding set of revoked users (we assume that $\mathcal{R}_t$ is part of RL_t). A revocable group signature (R-GS) scheme consists of the following algorithms or protocols.

$\mathsf{Setup}(\lambda, N)$: given a security parameter $\lambda \in \mathbb{N}$ and a maximal number of group members $N \in \mathbb{N}$, this algorithm (which is run by some trusted party) generates a group public key $\mathcal{Y}$, the group manager's private key $\mathcal{S}_{\mathsf{GM}}$ and the opening authority's private key $\mathcal{S}_{\mathsf{OA}}$. Keys $\mathcal{S}_{\mathsf{GM}}$ and $\mathcal{S}_{\mathsf{OA}}$ are given to the appropriate authority while $\mathcal{Y}$ is publicized. The algorithm also initializes a public state St comprising a set data structure $St_{users} = \emptyset$ and a string data structure $St_{\mathsf{trans}} = \epsilon$.

Join: is an interactive protocol between the group manager GM and a prospective group member $\mathcal{U}_i$. The protocol involves two interactive Turing machines $\mathsf{J}_{\mathsf{user}}$ and J_{GM} that both take as input $\mathcal{Y}$. The execution, denoted as $[\mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(\lambda, St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})]$, ends with $\mathcal{U}_i$ obtaining a membership secret sec_i, that no one else knows, and a membership certificate cert_i. If the protocol is successful, the group manager updates the public state St by setting $St_{users} := St_{users} \cup \{i\}$ as well as $St_{\mathsf{trans}} := St_{\mathsf{trans}} || \langle i, \mathsf{transcript}_i \rangle$.

Revoke: is a (possibly probabilistic) algorithm allowing the GM to generate an updated revocation list RL_t for the new revocation epoch t. It takes as input a public key $\mathcal{Y}$ and a set $\mathcal{R}_t \subset St_{users}$ that identifies the users to be revoked. It outputs an updated revocation list RL_t for epoch t.

Sign: given a revocation epoch t with its revocation list RL_t, a membership certificate cert_i, a membership secret sec_i and a message M, this algorithm outputs $\perp$ if $i \in \mathcal{R}_t$ and a signature σ otherwise.

Verify: given a signature σ, a revocation epoch t, the corresponding revocation list RL_t, a message M and a group public key $\mathcal{Y}$, this deterministic algorithm returns either 0 or 1.

Open: takes as input a message M, a valid signature σ w.r.t. $\mathcal{Y}$ for the indicated revocation epoch t, the opening authority's private key $\mathcal{S}_{\mathsf{OA}}$ and the public state St. It outputs $i \in St_{users} \cup \{\perp\}$, which is the identity of a group member or a symbol indicating an opening failure.

Each membership certificate contains a unique tag that identifies the user.

A R-GS scheme must satisfy three security notions defined in Appendix A. The first one is called *security against misidentification attacks*. It requires that, even if the adversary can introduce and revoke users at will, it cannot produce a signature that traces outside the set of unrevoked adversarially-controlled users.

As in ordinary (*i.e.*, non-revocable) group signatures, the notion of *security against framing attacks* captures that under no circumstances should an honest user be held accountable for messages that he did not sign, even if the whole system conspires against that user. Finally, the notion of *anonymity* is also defined (by granting the adversary access to a signature opening oracle) as in the models of [10, 40].

3 A Revocable Group Signature with Compact Keys and Constant Verification Time

The number of users is assumed to be $N = 2^{\ell-1} \in \mathsf{poly}(\lambda)$, for some integer ℓ, so that each group member is assigned to a leaf of the tree. Each node is assigned a unique identifier. For simplicity, the root is identified by $\mathsf{ID}(\epsilon) = 1$ and, for each other node x, we define the identifier $\mathsf{ID}(x) \in \{1, \ldots, 2N-1\}$ to be $\mathsf{ID}(x) = 2 \cdot \mathsf{ID}(\mathsf{parent}(x)) + b$, where $\mathsf{parent}(x)$ denotes x's father in the tree and $b = 0$ (resp. $b = 1$) if x is the left (resp. right) child of its father. The root of the tree is assigned the identifier $\mathsf{ID}_\epsilon = 1$.

At the beginning of each revocation epoch t, the GM generates an up-to-date revocation list RL_t containing one entry for each generic subset $S_{k_1,u_1}, \ldots, S_{k_m,u_m}$ produced by the Subset Difference method. These subsets are encoded in such a way that unrevoked users can anonymously prove their membership of one of them. Our technique allows to do this using a proof of *constant* size.

The intuition is as follows. In the generation of RL_t, for each $i \in \{1,\ldots,m\}$, if x_{k_i} (resp. x_{u_i}) denotes the primary (resp. secondary) root of S_{k_i,u_i}, the GM encodes S_{k_i,u_i} as a vector of group elements R_i that determines the levels of nodes x_{k_i} and x_{u_i} in the tree (which are called ϕ_i and ψ_i hereafter) and the identifiers $\mathsf{ID}(x_{k_i})$ and $\mathsf{ID}(x_{u_i})$. Then, the resulting vector R_i is authenticated by means of a structure preserving signature Θ_i, which is included in RL_t and will be used in a set membership proof [23].

During the join protocol, users obtain from the GM a structure-preserving signature on a compact encoding C_v – which is computed as a commitment to a vector of node identifiers $(I_1,\ldots,I_\ell)$ – of the path $(I_1,\ldots,I_\ell)$ between their leaf v and the root ϵ. This path is encoded as a single group element.

In order to anonymously prove his non-revocation, a group member $\mathcal{U}_i$ uses RL_t to determine the generic subset S_{k_l,u_l}, with $l \in \{1,\ldots,m\}$, where his leaf v_i lies. He commits to the corresponding vector of group elements R_l that encodes the node identifiers $\mathsf{ID}(x_{k_l})$ and $\mathsf{ID}(x_{u_l})$ of the primary and secondary roots of S_{k_l,u_l} at levels ϕ_l and ψ_l, respectively. If $(I_1,\ldots,I_\ell)$ identifies the path from his leaf v_i to ϵ, the unrevoked member $\mathcal{U}_i$ generates a membership proof for the subset S_{k_l,u_l} by proving that $\mathsf{ID}(x_{k_l}) = I_{\phi_l}$ and $\mathsf{ID}(x_{u_l}) \neq I_{\psi_l}$ (in other words, that x_{k_l} is an ancestor of v_i and x_{u_l} is not). To succinctly prove these statements, $\mathcal{U}_i$ uses the properties of the vector commitment scheme recalled in Section 2.4. Finally, in order to convince the verifier that he used a legal element of RL_t, $\mathcal{U}_i$ follows the technique of [23] and proves knowledge of a signature Θ_l on the committed vector of group elements R_l. By doing so, $\mathcal{U}_i$ thus provides evidence that his leaf v_i is a member of some authorized subset S_{k_l,u_l} without revealing $l \in \{1,\ldots,m\}$.

In order to obtain the strongest flavor of anonymity (*i.e.*, where the adversary has access to a signature opening oracle), the scheme uses Kiltz's tag-based encryption scheme [42] as in Groth's construction [35]. In non-frameability concerns, the group member $\mathcal{U}_i$ also generates a weak Boneh-Boyen signature [12] (which yields a fully secure signature when combined with a one-time signature) using $x = \log_g(X)$, where $X \in \mathbb{G}$ is a group element certified by the GM and bound to the path $(I_1,\ldots,I_\ell)$ during the join protocol.

3.1 Construction

As in standard security models for group signatures, we assume that, before joining the group, user $\mathcal{U}_i$ chooses a long term key pair $(\mathsf{usk}[i],\mathsf{upk}[i])$ and registers it in some PKI.

Setup(λ, N)**:** given a security parameter $\lambda \in \mathbb{N}$ and the permitted number of users $N = 2^{\ell-1}$,

1. Choose bilinear groups $(\mathbb{G},\mathbb{G}_T)$ of prime order $p > 2^\lambda$, with a generator $g \xleftarrow{R} \mathbb{G}$.
2. Define $n_0 = 2$ and $n_1 = 5$. Generate two key pairs $(sk^{(0)}_{\mathsf{AHO}}, pk^{(0)}_{\mathsf{AHO}})$ and $(sk^{(1)}_{\mathsf{AHO}}, pk^{(1)}_{\mathsf{AHO}})$ for the AHO signature in order to sign messages of n_0 and n_1 group elements, respectively. These key pairs are

$$pk^{(d)}_{\mathsf{AHO}} = \Big(G_r^{(d)},\ H_r^{(d)},\ G_z^{(d)} = G_r^{\gamma_z^{(d)}},\ H_z^{(d)} = H_r^{\delta_z^{(d)}},\ \{G_i^{(d)} = G_r^{\gamma_i^{(d)}}, H_i^{(d)} = H_r^{\delta_i^{(d)}}\}_{i=1}^{n_d},\ A^{(d)},\ B^{(d)}\Big)$$

and $sk^{(d)}_{\mathsf{AHO}} = \big(\alpha_a^{(d)}, \alpha_b^{(d)}, \gamma_z^{(d)}, \delta_z^{(d)}, \{\gamma_i^{(d)}, \delta_i^{(d)}\}_{i=1}^{n_d}\big)$, where $d \in \{0,1\}$. These two schemes will be used to sign messages consisting of 2 and 5 group elements, respectively.

3. Generate a public key $ck = (g_1, \ldots, g_\ell, g_{\ell+2}, \ldots, g_{2\ell}) \in \mathbb{G}^{2\ell-1}$ for vectors of dimension ℓ in the vector commitment scheme recalled in section 2.4. The trapdoor $g_{\ell+1}$ is not needed and can be discarded.
4. As a CRS for the NIWI proof system, select vectors $\mathbf{f} = (\vec{f_1}, \vec{f_2}, \vec{f_3})$ s.t. $\vec{f_1} = (f_1, 1, g) \in \mathbb{G}^3$, $\vec{f_2} = (1, f_2, g) \in \mathbb{G}^3$, and $\vec{f_3} = \vec{f_1}^{\xi_1} \cdot \vec{f_2}^{\xi_2}$, with $f_1 = g^{\beta_1}, f_2 = g^{\beta_2} \overset{R}{\leftarrow} \mathbb{G}$ and $\beta_1, \beta_2, \xi_1, \xi_2 \overset{R}{\leftarrow} \mathbb{Z}_p^*$. We also define the vector $\vec{\varphi} = \vec{f_3} \cdot (1, 1, g)$.
5. Choose $(U, V) \overset{R}{\leftarrow} \mathbb{G}^2$ that, together with generators $f_1, f_2, g \in \mathbb{G}$, will form a public encryption key.
6. Select a strongly unforgeable one-time signature $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$.
7. Set $\mathcal{S}_{\mathsf{GM}} := \big(sk_{\mathsf{AHO}}^{(0)}, sk_{\mathsf{AHO}}^{(1)}\big)$, $\mathcal{S}_{\mathsf{OA}} := \big(\beta_1, \beta_2\big)$ as authorities' private keys and the group public key is

$$\mathcal{Y} := \Big(g,\ pk_{\mathsf{AHO}}^{(0)},\ pk_{\mathsf{AHO}}^{(1)},\ ck = (g_1, \ldots, g_\ell, g_{\ell+2}, \ldots, g_{2\ell}),\ \mathbf{f},\ \vec{\varphi},\ (U, V),\ \Sigma\Big).$$

Join$^{(\mathrm{GM},\mathcal{U}_i)}$**:** the group manager and the prospective user $\mathcal{U}_i$ run the following interactive protocol $[\mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(\lambda, St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})]$:

1. $\mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y})$ draws $x \overset{R}{\leftarrow} \mathbb{Z}_p$ and computes $X = g^x$ which is sent to $\mathsf{J}_{\mathsf{GM}}(\lambda, St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})$. If $X \in \mathbb{G}$ already appears in some entry $\mathsf{transcript}_j$ of the database St_{trans}, J_{GM} halts and returns $\perp$ to $\mathsf{J}_{\mathsf{user}}$.
2. J_{GM} assigns to $\mathcal{U}_i$ an available leaf v of identifier $\mathsf{ID}(v)$ in the tree T. Let $x_1, \ldots, x_\ell$ be the path from $x_\ell = v$ to the root $x_1 = \epsilon$ of T. Let also $(I_1, \ldots, I_\ell) = (\mathsf{ID}(x_1), \ldots, \mathsf{ID}(x_\ell))$ be the corresponding vector of identifiers (with $I_1 = 1$ and $I_\ell = \mathsf{ID}(v) \in \{N, \ldots, 2N-1\}$). Then, J_{GM} does the following.
   a. Compute a compact encoding of $(I_1, \ldots, I_\ell)$ as $C_v = \prod_{\kappa=1}^{\ell} g_{\ell+1-\kappa}^{I_\kappa} = g_\ell^{I_1} \cdots g_1^{I_\ell}$.
   b. Using $sk_{\mathsf{AHO}}^{(0)}$, generate an AHO signature $\sigma_v = (\theta_{v,1}, \ldots, \theta_{v,7})$ on the pair $(X, C_v) \in \mathbb{G}^2$ so as to bind the encoded path C_v to the value X that identifies $\mathcal{U}_i$.
3. J_{GM} sends $\mathsf{ID}(v) \in \{N, \ldots, 2N-1\}$ and C_v to $\mathsf{J}_{\mathsf{user}}$ that halts if $\mathsf{ID}(v) \notin \{N, \ldots, 2N-1\}$ or if C_v is found incorrect. Otherwise, $\mathsf{J}_{\mathsf{user}}$ sends a signature $sig_i = \mathrm{Sign}_{\mathsf{usk}[i]}\big(X || (I_1, \ldots, I_\ell)\big)$ to J_{GM}.
4. J_{GM} checks that $\mathrm{Verify}_{\mathsf{upk}[i]}\big((X || (I_1, \ldots, I_\ell)), sig_i\big) = 1$. If not J_{GM} aborts. Otherwise, J_{GM} returns the AHO signature σ_v to $\mathsf{J}_{\mathsf{user}}$ and stores $\mathsf{transcript}_i = (X, \mathsf{ID}(v), C_v, \sigma_v, sig_i)$ in the database St_{trans}.
5. $\mathsf{J}_{\mathsf{user}}$ defines the membership certificate as $\mathsf{cert}_i = \big(\mathsf{ID}(v), X, C_v, \sigma_v\big) \in \{N, \ldots, 2N-1\} \times \mathbb{G}^9$, where X will serve as the tag identifying $\mathcal{U}_i$. The membership secret sec_i is defined as $\mathsf{sec}_i = x \in \mathbb{Z}_p$.

Revoke$(\mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, t, \mathcal{R}_t)$**:** Parse $\mathcal{S}_{\mathsf{GM}}$ as $\mathcal{S}_{\mathsf{GM}} := \big(sk_{\mathsf{AHO}}^{(0)}, sk_{\mathsf{AHO}}^{(1)}\big)$ and do the following.

1. Using the subset covering algorithm of the SD method, find a cover of the unrevoked user set $\{1, \ldots, N\} \backslash \mathcal{R}_t$ as the union of disjoint subsets of the form $S_{k_1,u_1}, \ldots, S_{k_m,u_m}$, where $m \leq 2 \cdot |\mathcal{R}_t| - 1$.
2. For $i = 1$ to m, do the following.
   a. Consider the subset S_{k_i,u_i} as the difference between sub-trees rooted at an internal node x_{k_i} and one of its descendants x_{u_i}. Let $\phi_i, \psi_i \in \{1, \ldots, \ell\}$ be the depths of x_{k_i} and

x_{u_i}, respectively, in T assuming that the root ϵ is at depth 1. Encode S_{k_i,u_i} as a vector $\big(g_{\phi_i}, g_1^{\mathsf{ID}(x_{k_i})}, g_{\psi_i}, g^{\mathsf{ID}(x_{u_i})}\big)$.

b. To authenticate S_{k_i,u_i} and bind it to the revocation epoch t, use $sk_{\mathsf{AHO}}^{(1)}$ to generate an AHO signature $\Theta_i = (\Theta_{i,1}, \ldots, \Theta_{i,7}) \in \mathbb{G}^7$ on $R_i = \Big(g^t, g_{\phi_i}, g_1^{\mathsf{ID}(x_{k_i})}, g_{\psi_i}, g^{\mathsf{ID}(x_{u_i})}\Big)$, where the epoch number t is interpreted as an element of $\mathbb{Z}_p$.

Return the revocation data

$$RL_t = \Big(t,\ \mathcal{R}_t,\ \{\phi_i,\ \psi_i,\ \mathsf{ID}(x_{k_i}),\ \mathsf{ID}(x_{u_i}),\ \Theta_i = (\Theta_{i,1}, \ldots, \Theta_{i,7})\}_{i=1}^m\Big). \quad (4)$$

Sign$(\mathcal{Y}, t, RL_t, \mathsf{cert}_i, \mathsf{sec}_i, M)$**:** return $\perp$ if $i \in \mathcal{R}_t$. Otherwise, to sign $M \in \{0,1\}^*$, generate a one-time signature key pair $(\mathsf{SK}, \mathsf{VK}) \leftarrow \mathcal{G}(\lambda)$. Parse cert_i as $\mathsf{cert}_i = \big(\mathsf{ID}(v_i), X, C_{v_i}, \sigma_{v_i}\big) \in \{N, \ldots, 2N-1\} \times \mathbb{G}^9$ and sec_i as $x \in \mathbb{Z}_p$. Let $\epsilon = x_1, \ldots, x_\ell = v_i$ be the path connecting v_i to the root ϵ of T and let $(I_1, \ldots, I_\ell) = (\mathsf{ID}(x_1), \ldots, \mathsf{ID}(x_\ell))$ be the vector of node identifiers. First, $\mathcal{U}_i$ generates a commitment $com_{C_{v_i}}$ to the encoding C_{v_i} of the path $(I_1, \ldots, I_\ell)$ from v_i to the root. Then, he does the following.

1. Using RL_t, find the set S_{k_l,u_l}, with $l \in \{1, \ldots, m\}$, that contains the leaf v_i identified by $\mathsf{ID}(v_i)$. Let x_{k_l} and x_{u_l} denote the primary and secondary roots of S_{k_l,u_l} at depths ϕ_l and ψ_l, respectively. Since x_{k_l} is an ancestor of v_i but x_{u_l} is not, it must be the case that $I_{\phi_l} = \mathsf{ID}(x_{k_l})$ and $I_{\psi_l} \neq \mathsf{ID}(x_{u_l})$.

2. To prove that v_i belongs to S_{k_l,u_l} without leaking l, $\mathcal{U}_i$ first re-randomizes the l-th AHO signature Θ_l of RL_t as $\{\Theta'_{l,i}\}_{i=1}^7 \leftarrow \mathsf{ReRand}(pk_{\mathsf{AHO}}^{(1)}, \Theta_l)$. Then, he commits to the l-th revocation message

$$R_l = (R_{l,1}, R_{l,2}, R_{l,3}, R_{l,4}, R_{l,5}) = \big(g^t, g_{\phi_l}, g_1^{\mathsf{ID}(x_{k_l})}, g_{\psi_l}, g^{\mathsf{ID}(x_{u_l})}\big) \quad (5)$$

and its signature $\Theta'_l = (\Theta'_{l,1}, \ldots, \Theta'_{l,7})$ by computing Groth-Sahai commitments $\{com_{R_{l,\tau}}\}_{\tau=2}^5$, $\{com_{\Theta'_{l,j}}\}_{j\in\{1,2,5\}}$ to $\{R_{l,\tau}\}_{\tau=2}^5$ and $\{\Theta'_{l,j}\}_{j\in\{1,2,5\}}$.

a. To prove that $I_{\phi_l} = \mathsf{ID}(x_{k_l})$, $\mathcal{U}_i$ first computes $W_{\phi_l} = \prod_{\kappa=1,\ \kappa\neq\phi_l}^{\ell} g_{\ell+1-\kappa+\phi_l}^{I_\kappa}$ that satisfies the equality $e(g_{\phi_l}, C_{v_i}) = e(g_1, g_\ell)^{I_{\phi_l}} \cdot e(g, W_{\phi_l})$. Then, $\mathcal{U}_i$ generates a Groth-Sahai commitment $com_{W_{\phi_l}}$ to W_{ϕ_l}. He computes a NIWI proof that committed variables $(R_{l,2}, R_{l,3}, C_{v_i}, W_{\phi_l})$ satisfy

$$e(R_{l,2}, C_{v_i}) = e(R_{l,3}, g_\ell) \cdot e(g, W_{\phi_l}). \quad (6)$$

We denote by π_{eq} the proof for the quadratic equation (6), which requires 9 group elements.

b. To prove that $I_{\psi_l} \neq \mathsf{ID}(x_{u_l})$, $\mathcal{U}_i$ computes $W_{\psi_l} = \prod_{\kappa=1,\ \kappa\neq\psi_l}^{\ell} g_{\ell+1-\kappa+\psi_l}^{I_\kappa}$ that satisfies the equality $e(g_{\psi_l}, C_{v_i}) = e(g_1, g_\ell)^{I_{\psi_l}} \cdot e(g, W_{\psi_l})$. Then, he computes a commitment $com_{W_{\psi_l}}$ to W_{ψ_l} as well as commitments com_{Γ_l} and $\{com_{\Psi_{l,\tau}}\}_{\tau\in\{0,1,2\ell\}}$ to the group elements

$$(\Gamma_l, \Psi_{l,0}, \Psi_{l,1}, \Psi_{l,2\ell}) = \big(g^{1/(I_{\psi_l} - \mathsf{ID}(x_{u_l}))}, g^{I_{\psi_l}}, g_1^{I_{\psi_l}}, g_{2\ell}^{I_{\psi_l}}\big).$$

Then, $\mathcal{U}_i$ provides evidence that committed variables $(R_{l,4}, R_{l,5}, C_{v_i}, \Gamma_l, \Psi_{l,0}, \Psi_{l,1}, \Psi_{l,2\ell})$ satisfy

$$e(R_{l,4}, C_{v_i}) = e(\Psi_{l,1}, g_\ell) \cdot e(g, W_{\psi_l}), \qquad e(\Psi_{l,0}/R_{l,5}, \Gamma_l) = e(g,g) \tag{7}$$
$$e(\Psi_{l,1}, g) = e(g_1, \Psi_{l,0}), \qquad e(\Psi_{l,2\ell}, g) = e(g_{2\ell}, \Psi_{l,0}). \tag{8}$$

We denote this NIWI proof by $\pi_{neq} = (\pi_{neq,1}, \pi_{neq,2}, \pi_{neq,3}, \pi_{neq,4})$. Since the first two equations (7) are quadratic, $\pi_{neq,1}$ and $\pi_{neq,2}$ consist of 9 elements each. The last two equations (8) are linear and both cost 3 elements to prove.

3. $\mathcal{U}_i$ provides evidence that the tuple R_l of (5) is a certified revocation message for epoch t: namely, he computes a NIWI proof π_{R_l} that committed message elements $\{R_{l,\tau}\}_{\tau=2}^{5}$ and signature components $\{\Theta'_{l,j}\}_{j\in\{1,2,5\}}$ satisfy the equations

$$A^{(1)} \cdot e(\Theta'_{l,3}, \Theta'_{l,4})^{-1} \cdot e(G_1^{(1)}, g^t)^{-1} = e(G_z^{(1)}, \Theta'_{l,1}) \cdot e(G_r^{(1)}, \Theta'_{l,2}) \cdot \prod_{\tau=2}^{5} e(G_\tau^{(1)}, R_{l,\tau}), \tag{9}$$
$$B^{(1)} \cdot e(\Theta'_{l,6}, \Theta'_{l,7})^{-1} \cdot e(H_1^{(1)}, g^t)^{-1} = e(H_z^{(1)}, \Theta'_{l,1}) \cdot e(H_r^{(1)}, \Theta'_{l,5}) \cdot \prod_{\tau=2}^{5} e(H_\tau^{(1)}, R_{l,\tau}),$$

Since $\{\Theta'_{l,j}\}_{j\in\{3,4,6,7\}}$ are constants, equations (9) are both linear and thus require 3 elements each. Hence, π_{R_l} takes 6 elements altogether.

4. Let $\sigma_{v_i} = (\theta_{v_i,1}, \ldots, \theta_{v_i,7})$ be the AHO signature on the message (X, C_{v_i}). Set $\{\theta'_{v_i,j}\}_{j=1}^{7} \leftarrow \mathsf{ReRand}(pk_{\mathsf{AHO}}^{(0)}, \sigma_{v_i})$ and generate commitments $\{com_{\theta'_{v_i,j}}\}_{j\in\{1,2,5\}}$ to $\{\theta'_{v_i,j}\}_{j\in\{1,2,5\}}$ as well as a commitment com_X to X. Then, generate a NIWI proof $\pi_{\sigma_{v_i}}$ that committed variables satisfy the verification equations

$$A^{(0)} \cdot e(\theta'_{l,3}, \theta'_{l,4})^{-1} = e(G_z^{(0)}, \theta'_{l,1}) \cdot e(G_r^{(0)}, \theta'_{l,2}) \cdot e(G_1^{(0)}, X) \cdot e(G_2^{(0)}, C_{v_i}),$$
$$B^{(0)} \cdot e(\theta'_{l,6}, \theta'_{l,7})^{-1} = e(H_z^{(0)}, \theta_{l,1}) \cdot e(H_r^{(0)}, \theta'_{l,5}) \cdot e(H_1^{(0)}, X) \cdot e(H_2^{(0)}, C_{v_i})$$

Since these equations are linear, $\pi_{\sigma_{v_i}}$ requires 6 group elements.

5. Using VK as a tag (by first hashing it onto $\mathbb{Z}_p$ in such a way that it can be interpreted as a $\mathbb{Z}_p$ element), compute a tag-based encryption [42] of X by drawing $z_1, z_2 \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and setting $(\Upsilon_1, \Upsilon_2, \Upsilon_3, \Upsilon_4, \Upsilon_5) = \big(f_1^{z_1}, f_2^{z_2}, X \cdot g^{z_1+z_2}, (g^{\mathsf{VK}} \cdot U)^{z_1}, (g^{\mathsf{VK}} \cdot V)^{z_2}\big)$.

6. Generate a NIZK proof that $com_X = (1,1,X) \cdot \vec{f_1}^{\,w_{X,1}} \cdot \vec{f_2}^{\,w_{X,2}} \cdot \vec{f_3}^{\,w_{X,3}}$ and $(\Upsilon_1, \Upsilon_2, \Upsilon_3)$ are BBS encryptions of the same value X. If we write $\vec{f_3} = (f_{3,1}, f_{3,2}, f_{3,3})$, the Groth-Sahai commitment com_X can be written as $(f_1^{w_{X,1}} \cdot f_{3,1}^{w_{X,3}}, f_2^{w_{X,2}} \cdot f_{3,2}^{w_{X,3}}, X \cdot g^{w_{X,1}+w_{X,2}} \cdot f_{3,3}^{w_{X,3}})$, so that we have

$$com_X \cdot (\Upsilon_1, \Upsilon_2, \Upsilon_3)^{-1} = \big(f_1^{\chi_1} \cdot f_{3,1}^{\chi_3},\; f_2^{\chi_2} \cdot f_{3,2}^{\chi_3},\; g^{\chi_1+\chi_2} \cdot f_{3,3}^{\chi_3}\big) \tag{10}$$

with $\chi_1 = w_{X,1} - z_1$, $\chi_2 = w_{X,2} - z_2$, $\chi_3 = w_{X,3}$. Compute $com_{\chi_j} = \vec{\varphi}^{\,\chi_j} \cdot \vec{f_1}^{\,w_{\chi_j,1}} \cdot \vec{f_2}^{\,w_{\chi_j,2}}$, with $w_{\chi_j,1}, w_{\chi_j,2} \stackrel{R}{\leftarrow} \mathbb{Z}_p$ for $j \in \{1,2,3\}$, as commitments to $\{\chi_j\}_{j=1}^{3}$ and generates proofs $\{\pi_{eq\text{-}com,j}\}_{j=1}^{3}$ that χ_1, χ_2, χ_3 satisfy the three linear relations (10). The proofs $\{\pi_{eq\text{-}com,j}\}_{j=1}^{3}$ cost 2 elements each.

7. Compute a weak Boneh-Boyen signature $\sigma_{\mathsf{VK}} = g^{1/(x+\mathsf{VK})}$ on VK and a commitment $com_{\sigma_{\mathsf{VK}}}$ to σ_{VK}. Then, generate a NIWI proof $\pi_{\sigma_{\mathsf{VK}}} = (\vec{\pi}_{\sigma_{\mathsf{VK}},1}, \vec{\pi}_{\sigma_{\mathsf{VK}},2}, \vec{\pi}_{\sigma_{\mathsf{VK}},3}) \in \mathbb{G}^9$ that committed variables $(\sigma_{\mathsf{VK}}, X) \in \mathbb{G}^2$ satisfy the quadratic equation $e(\sigma_{\mathsf{VK}}, X \cdot g^{\mathsf{VK}}) = e(g, g)$.
8. Compute $\sigma_{ots} = \mathcal{S}(\mathsf{SK}, (M, RL_t, \Upsilon_1, \Upsilon_2, \Upsilon_3, \Upsilon_4, \Upsilon_5, \Omega, \mathbf{com}, \mathbf{\Pi}))$ where $\Omega = \{\Theta'_{l,i}, \theta'_{l,i}\}_{i\in\{3,4,6,7\}}$ and

$$\mathbf{com} = \big(com_{C_{v_i}}, com_X, \{com_{R_{l,\tau}}\}_{\tau=2}^{5}, com_{W_{\phi_l}}, com_{W_{\psi_l}}, com_{\Gamma_l}, \{com_{\Psi_{l,\tau}}\}_{\tau\in\{0,1,2\ell\}},$$
$$\{com_{\Theta'_{l,j}}\}_{j\in\{1,2,5\}}, \{com_{\theta'_{l,j}}\}_{j\in\{1,2,5\}}, \{com_{\chi_j}\}_{j=1}^{3}, com_{\sigma_{\mathsf{VK}}}\big)$$
$$\mathbf{\Pi} = \big(\pi_{eq}, \pi_{neq}, \pi_{R_l}, \pi_{\sigma_{v_i}}, \{\pi_{eq\text{-}com,j}\}_{j=1}^{3}, , \pi_{\sigma_{\mathsf{VK}}}\big)$$

Return the signature $\sigma = \big(\mathsf{VK}, \Upsilon_1, \Upsilon_2, \Upsilon_3, \Upsilon_4, \Upsilon_5, \Omega, \mathbf{com}, \mathbf{\Pi}, \sigma_{ots}\big)$.

Verify$(\sigma, M, t, RL_t, \mathcal{Y})$**:** parse σ as above. If $\mathcal{V}(\mathsf{VK}, (M, RL_t, \Upsilon_1, \Upsilon_2, \Upsilon_3, \Upsilon_4, \Upsilon_5, \Omega, \mathbf{com}, \mathbf{\Pi}), \sigma_{ots}) = 0$ or if $(\Upsilon_1, \Upsilon_2, \Upsilon_3, \Upsilon_4, \Upsilon_5)$ is not a well-formed tag-based encryption (that is, if $e(\Upsilon_1, g^{\mathsf{VK}} \cdot U) \neq e(f_1, \Upsilon_4)$ or $e(\Upsilon_2, g^{\mathsf{VK}} \cdot V) \neq e(f_2, \Upsilon_5)$), return 0. Then, return 1 if all proofs properly verify. Otherwise, return 0.

Open$(M, t, RL_t, \sigma, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, St)$**:** parse σ as above and return $\perp$ if $\mathsf{Verify}(\sigma, M, t, RL_t, \mathcal{Y}) = 0$. Otherwise, given $\mathcal{S}_{\mathsf{OA}} = (\beta_1, \beta_2)$, compute $\tilde{X} = \Upsilon_3 \cdot \Upsilon_1^{-1/\beta_1} \cdot \Upsilon_2^{-1/\beta_2}$. In the database St_{trans}, find a record $\langle i, \mathsf{transcript}_i = (X_i, \mathsf{ID}(v_i), C_{v_i}, \sigma_{v_i}, sig_i)\rangle$ such that $X_i = \tilde{X}$. If no such record exists in St_{trans}, return $\perp$. Otherwise, return i.

At first glance, the variable $\Psi_{l,2\ell}$ and the proof of the second equality (8) may seem unnecessary in step 2.b of the signing algorithm. However, this element plays a crucial role when it comes to prove the security under the ℓ-FlexDHE assumption. Indeed, the proof of security against misidentification attacks (more precisely, the proof of Lemma 1 in Appendix B.1) ceases to go through if we remove $\Psi_{l,2\ell}$ and its corresponding proof.

As far as efficiency goes, each entry of RL_t contains 7 group elements and two node identifiers of $O(\log N)$ bits each. If $\lambda_{\mathbb{G}}$ is the bitlength of a group element, we have $\log N \ll \lambda_{\mathbb{G}}/2$ (since $\lambda \leq \lambda_{\mathbb{G}}$ and N is polynomial), so that the number of bits of RL_t is bounded by $2 \cdot |\mathcal{R}_t| \cdot (7 \cdot \lambda_{\mathbb{G}} + 2\log N + 2\log\log N) < 2 \cdot |\mathcal{R}_t| \cdot (9\lambda_{\mathbb{G}})$ bits. The size of RL_t is thus bounded by that of $18 \cdot |\mathcal{R}_t|$ group elements.

Unlike [47], group members only need to store 9 group elements in their membership certificate. As far as the size of signature goes, **com** and $\mathbf{\Pi}$ require 66 and 60 group elements, respectively. If the one-time signature of [34] is used, VK and σ_{ots} consist of 3 elements of $\mathbb{G}$ and 2 elements of $\mathbb{Z}_p$, respectively. The global size σ amounts to that of 144 group elements, which is about 50% longer than [47]. In comparison with [35] (which does not natively support revocation), signatures are only longer by a factor of 3. At the 128-bit security level, each group element should have a 512-bit representation and a signature takes 9 kB.

Verifying signatures takes constant time. The signer has to compute at most $2\ell = O(\log N)$ exponentiations to obtain W_{ϕ_l} and W_{ψ_l} at the beginning of each revocation epoch. Note that these exponentiations involve short exponents of $O(\log N)$ bits each. Hence, computing W_{ϕ_l} and W_{ψ_l} requires $O(\log^2 N)$ multiplications in $\mathbb{G}$. For this reason, since we always have $\log^2 N \ll \lambda$ (as long as $N \ll 2^{\lambda^{1/2}}$), this cost is dominated by that of a single exponentiation in $\mathbb{G}$.

3.2 Security

From a security point of view, we prove the following theorem in Appendix B.

Theorem 1. *The scheme provides anonymity as well as security against misidentification and framing attacks if the SFP, FlexDHE, SDH and DLIN assumptions all hold in $\mathbb{G}$.*

In comparison with [47], the security proof requires the additional non-standard ℓ-FlexDHE assumption, where $\ell = \log(N)$. In Appendix C, we show how to rest on weaker (and fewer) intractability assumptions if we accept to use a group public key of size $O(\log^2 N)$ while keeping all other complexities unchanged. This construction offers an interesting tradeoff since, in some applications, group public keys of log-squared size are handier to work with than private keys of size $O(\log^3 N)$ as in [47].

Appendix C also explains how to also eliminate the SDH assumption using the technique of Malkin et al. [48]. In this case, an additive factor of $O(\lambda)$ appears in the group public key size because a longer Groth-Sahai CRS must be used. On the other hand, the q-SFP assumption becomes the only assumption of variable size.

4 Efficiency Comparisons

This section compares pairing-based revocable group signatures where group members are stateless and do not update their membership certificate whenever a revocation occurs. Comparisons are given in terms of computational costs and the size (measured by the number of group elements) of public keys, signatures, membership certificates and revocation lists as functions of N, r and, in some cases, the number T of revocation epochs. By "constant", we thus mean that the complexity only depends on the security parameter λ.

Table 1. Comparison between pairing-based revocable group signatures

Schemes	Group public	Signature	Membership	Revocation	Signature	Verification	Revocation	Standard
	key size	size	certificate size	list size	cost	cost	cost	model?
NFHF1 [49]	$O(N)$	$O(1)$	$O(1)$	$O(r)$	$O(1)$	$O(1)$	$O(r)$	✗
NFHF2 [49]	$O(N^{1/2})$	$O(1)$	$O(1)$	$O(r)$	$O(1)$	$O(1)$	$O(r)$	✗
BS [17]	$O(1)$	$O(1)$	$O(1)$	$O(r)$	$O(1)$	$O(r)$	$O(1)$	✗
NF [50]	$O(T)^{\diamondsuit}$	$O(1)$	$O(1)$	$O(r)$	$O(1)$	$O(r)$	$O(r)$	✗
LV [45]	$O(T)^{\diamondsuit}$	$O(1)$	$O(1)$	$O(r)$	$O(1)$	$O(r)$	$O(r)$	✓
LPY1 (SD)	$O(\log N)$	$O(1)$	$O(\log^3 N)$	$O(r)$	$O(\log N)^{\dagger}$	$O(1)$	$O(r \cdot \log N)$	✓
LPY2 (CS)	$O(1)$	$O(1)$	$O(\log N)$	$O(r \cdot \log(N/r))$	$O(1)$	$O(1)$	$O(r \cdot \log(N/r))$	✓
This work	$O(\log N)$	$O(1)$	$O(1)$	$O(r)$	$O(1)$	$O(1)$	$O(r)$	✓

N: max. number of users; r: number of revocations T: max. number of revocation epochs

$\diamondsuit$ These schemes can be modified to have $O(1)$-size group public keys.

$\dagger$ This complexity is only involved at the first signature of each revocation epoch.

As previously mentioned, among schemes where revocations require no update in unrevoked users' credentials, the new method seems asymptotically optimal. The only dependency on N appears in the group public key size, which is logarithmic and thus quite moderate. At the same time, it retains revocation lists of size $O(r)$ (which is on par with the VLR-based approach [17] but without its verification cost of $O(r)$) as in the SD method of [47]. In comparison with the latter, we also eliminate the $O(\log N)$ multiplicative factor in the revocation cost and the complexity of the signing algorithm in the worst case.

The joining protocol is also much more efficient in our scheme than in [47] as the group manager has to generate only one structure-preserving signature (computing C_v in step 2.a of the protocol is actually cheaper than a single exponentiation in $\mathbb{G}$), instead of $\log(N)$ in the two schemes of [47].

In Appendix C, we give tradeoffs between the strength of the assumption and the efficiency: in these alternative constructions, the assumption is weakened at the expense of group public keys of size $O(\log^2 N)$ or $O(\lambda + \log^2 N)$.

References

1. M. Abe, K. Haralambiev, M. Ohkubo. Signing on Elements in Bilinear Groups for Modular Protocol Design. Cryptology ePrint Archive: Report 2010/133, 2010.
2. M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, M. Ohkubo. Structure-Preserving Signatures and Commitments to Group Elements. In *Crypto'10*, *LNCS* 6223, pp. 209–236, 2010.
3. M. Abe, J. Groth, K. Haralambiev, M. Ohkubo. Optimal Structure-Preserving Signatures in Asymmetric Bilinear Groups. In *Crypto'11*, *LNCS* 6841, pp. 649–666, 2011.
4. T. Acar, L. Nguyen. Revocation for Delegatable Anonymous Credentials. In *PKC'11*, *LNCS* 6571, pp. 423–440, 2011.
5. G. Ateniese, J. Camenisch, S. Hohenberger, B. de Medeiros. Practical group signatures without random oracles. Cryptology ePrint Archive: Report 2005/385, 2005.
6. G. Ateniese, J. Camenisch, M. Joye, G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *Crypto'00*, *LNCS* 1880, pp. 255–270, 2000.
7. G. Ateniese, D. Song, G. Tsudik. Quasi-Efficient Revocation in Group Signatures. In *Financial Cryptography'02*, *LNCS* 2357, pp. 183–197, 2002.
8. M. Bellare, D. Micciancio, B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *Eurocrypt'03*, *LNCS* 2656, pp. 614–629, 2003.
9. M. Bellare, P. Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *1st ACM Conference on Computer and Communications Security*, pp. 62–73, ACM Press, 1993.
10. M. Bellare, H. Shi, C. Zhang. Foundations of group signatures: The case of dynamic groups. In *CT-RSA'05*, *LNCS* 3376, pp. 136–153, 2005.
11. J. Benaloh, M. de Mare. One-Way Accumulators: A Decentralized Alternative to Digital Sinatures. In *Eurocrypt'93*, *LNCS* 4948, pp. 274–285, 1993.
12. D. Boneh, X. Boyen. Short Signatures Without Random Oracles. In *Eurocrypt'04*, *LNCS* 3027, pp. 56–73. Springer-Verlag, 2004.
13. D. Boneh, X. Boyen, E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In *Eurocrypt'05*, *LNCS* 3494, pp. 440–456, 2005.
14. D. Boneh, X. Boyen, H. Shacham. Short Group Signatures. In *Crypto'04*, *LNCS* 3152, pp. 41–55. Springer, 2004.
15. D. Boneh and M. Franklin. Identity based encryption from the Weil pairing. *SIAM J. of Computing*, 32(3):586–615, 2003. Extended abstract in Crypto'01, LNCS 2139, pp. 213–229, 2001.
16. D. Boneh, C. Gentry and B. Waters. Collusion-Resistant Broadcast Encryption with Short Ciphertexts and Private Keys. In *Crypto'05*, *LNCS* 3621, pp. 258–275, 2005.
17. D. Boneh, H. Shacham. Group signatures with verifier-local revocation. In *ACM-CCS'04*, pp. 168–177. ACM Press, 2004.
18. X. Boyen, B. Waters. Compact Group Signatures Without Random Oracles. In *Eurocrypt'06*, *LNCS* 4004, pp. 427–444, Springer, 2006.
19. X. Boyen, B. Waters. Full-Domain Subgroup Hiding and Constant-Size Group Signatures. In *PKC'07*, *LNCS* 4450, pp. 1–15, 2007.
20. E. Bresson, J. Stern. Efficient Revocation in Group Signatures. In *PKC'01*, *LNCS* 1992, pp. 190–206, 2001.
21. E. Brickell. An efficient protocol for anonymously providing assurance of the container of the private key. Submission to the Trusted Computing Group. April, 2003.
22. E. Brickell, J. Camenisch, L. Chen. Direct Anonymous Attestation. In *ACM-CCS'04*, pp. 132–145, 2004.
23. J. Camenisch, R. Chaabouni, a. shelat. Efficient Protocols for Set Membership and Range Proofs. In *Asiacrypt'08*, *LNCS* 5350, pp. 234–252, Springer, 2008.
24. J. Camenisch, M. Kohlweiss, C. Soriente. An Accumulator Based on Bilinear Maps and Efficient Revocation for Anonymous Credentials. In *PKC'09*, *LNCS* 5443, pp. 481–500, 2009.

25. J. Camenisch, M. Kohlweiss, C. Soriente. Solving Revocation with Efficient Update of Anonymous Credentials. In *SCN'10*, *LNCS* 6280, pp. 454–471, 2010.
26. J. Camenisch, A. Lysyanskaya. Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. In *Crypto'02*, *LNCS* 2442, pp. 61–76, Springer, 2002.
27. D. Catalano, D. Fiore. Concise Vector Commitments and their Applications to Zero-Knowledge Elementary Databases. In Cryptology ePrint Archive: Report 2011/495, 2011.
28. J. Cathalo, B. Libert, M. Yung. Group Encryption: Non-Interactive Realization in the Standard Model. In *Asiacrypt'09*, *LNCS* 5912, pp. 179–196, 2009.
29. D. Chaum, E. van Heyst. Group Signatures. In *Eurocrypt'91*, *LNCS* 547, pp. 257–265, Springer, 1991.
30. C. Delerablée, D. Pointcheval. Dynamic Fully Anonymous Short Group Signatures. In *Vietcrypt'06*, *LNCS* 4341, pp. 193–210, Springer, 2006.
31. Y. Dodis, N. Fazio. Public Key Broadcast Encryption for Stateless Receivers. In *Digital Rights Management (DRM'02)*, *LNCS* 2696, pp. 61–80, 2002.
32. G. Fuchsbauer. Automorphic Signatures in Bilinear Groups and an Application to Round-Optimal Blind Signatures. Cryptology ePrint Archive: Report 2009/320, 2009.
33. C. Gentry, A. Silverberg. Hierarchical ID-based cryptography. In *Asiacrypt'02*, *LNCS* 2501, Springer, 2002.
34. J. Groth. Simulation-Sound NIZK Proofs for a Practical Language and Constant Size Group Signatures. In *Asiacrypt'06*, *LNCS* 4284, pp. 444–459, Springer, 2006.
35. J. Groth. Fully anonymous group signatures without random oracles. In *Asiacrypt 2007*, *LNCS* 4833, pp. 164–180. Springer, 2007.
36. J. Groth, A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Eurocrypt'08*, LNCS 4965, pp. 415–432, 2008.
37. D. Halevy, A. Shamir. The LSD broadcast encryption scheme. In *Crypto'02*, *LNCS* 2442, pp. 47–60, Springer, 2002.
38. J. Horwitz, B. Lynn. Toward hierarchical identity-based encryption. In *Eurocrypt'02*, *LNCS* 2332, Springer, 2002.
39. M. Izabachène, B. Libert, D. Vergnaud. Blockwise P-Signatures and Non-Interactive Anonymous Credentials with Efficient Attributes. 13*th IMA International Conference on Cryptography and Coding (IMACC 2011)*, pp. 431–450, Springer, 2011.
40. A. Kiayias, M. Yung. Secure scalable group signature with dynamic joins and separable authorities. International Journal of Security and Networks (IJSN) Vol. 1, No. 1/2, pp. 24–45, 2006. Earlier version appeared as Cryptology ePrint Archive: Report 2004/076, 2004.
41. A. Kiayias, M. Yung. Group signatures with efficient concurrent join. In *Eurocrypt'05*, *LNCS* 3494, pp. 198–214, 2005.
42. E. Kiltz. Chosen-ciphertext security from tag-based encryption. In *TCC'06*, *LNCS* 3876, pp. 581–600, 2006.
43. S. Kunz-Jacques and D. Pointcheval. About the security of MTI/C0 and MQV. In *SCN'06*, LNCS 4116, pages 156–172, 2006.
44. F. Laguillaumie, P. Paillier, D. Vergnaud. Universally Convertible Directed Signatures. In *Asiacrypt'05*, *LNCS* 3788, pp. 682–701, 2005.
45. B. Libert, D. Vergnaud. Group Signatures with Verifier-Local Revocation and Backward Unlinkability in the Standard Model. In *CANS'09*, *LNCS* 5888, pp. 498-517, 2009.
46. B. Libert and M. Yung. Concise Mercurial Vector Commitments and Independent Zero-Knowledge Sets with Short Proofs. In *TCC'10*, LNCS 5978, pp. 499–517, 2010.
47. B. Libert, T. Peters and M. Yung. Scalable Group Signatures with Revocation. In *Eurocrypt'12*, *LNCS* series, to appear, 2012.
48. T. Malkin, I. Teranishi, Y. Vahlis, M. Yung. Signatures resilient to continual leakage on memory and computation. In *TCC'11*, *LNCS* 6597, pp. 89–106, 2011.
49. T. Nakanishi, H. Fujii, Y. Hira, N. Funabiki. Revocable Group Signature Schemes with Constant Costs for Signing and Verifying. In *PKC'09*, *LNCS* 5443, pp. 463–480, 2009.
50. T. Nakanishi, N. Funabiki. Verifier-Local Revocation Group Signature Schemes with Backward Unlinkability from Bilinear Maps. In *Asiacrypt'05*, *LNCS* 5443, pp. 533–548, 2009.
51. M. Naor, D. Naor, J. Lotspiech. Revocation and Tracing Schemes for Stateless Receivers. In *Crypto'01*, *LNCS* 2139, pp. 41–62, 2001.
52. M. Naor. On Cryptographic Assumptions and Challenges. In *Crypto'03*, *LNCS* 2729, pp. 96–109. Springer-Verlag, 2003.
53. L. Nguyen. Accumulators from Bilinear Pairings and Applications. In *CT-RSA'05*, *LNCS* 3376, pp. 275–292, 2005.

54. L. Nguyen, R. Safavi-Naini. Efficient and Provably Secure Trapdoor-Free Group Signature Schemes from Bilinear Pairings. In *Asiacrypt'04*, *LNCS* 3329, pp. 372–386. Springer-Verlag, 2004.
55. V. Shoup. Lower bounds for discrete logarithms and related problems. In *Eurocrypt'97*, *LNCS* 1233, pp. 256–66, 1997.
56. D. Song. Practical forward secure group signature schemes. In *ACM-CCS'01*, pp. 225–234, 2001.
57. P. Tsang, M.-Ho Au, A. Kapadia, S. Smith. Blacklistable anonymous credentials: blocking misbehaving users without TTPs. In *ACM-CCS'07*, pp. 72–81, 2007.
58. P. Tsang, M.-Ho Au, A. Kapadia, S. Smith. PEREA: towards practical TTP-free revocation in anonymous authentication. In *ACM-CCS'08*, pp. 333–344, 2008.
59. G. Tsudik, S. Xu. Accumulating Composites and Improved Group Signing. In *Asiacrypt'03*, *LNCS* 2894, pp. 269–286, 2003.
60. B. Waters. Efficient identity-based encryption without random oracles. In *Eurocrypt 2005*, *LNCS* 2567. Springer, 2005.
61. S. Zhou, D. Lin. Shorter Verifier-Local Revocation Group Signatures from Bilinear Maps. In *CANS'06*, *LNCS* 4301, pp. 126–143, Springer, 2006.

A Correctness and Security Definitions for Revocable Group Signatures

In the following, as in [40], we say that a public state St is *valid* if it is reachable from $St = (\emptyset, \varepsilon)$ by a Turing machine having oracle access to J_{GM}. Also, a state St' is said to *extend* another state St if it is within reach from St.

As in [40, 41], when we write $\mathsf{cert}_i \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_i$, it means that there exist coin tosses ϖ for J_{GM} and J_{user} such that, for some valid public state St', the execution of $[\mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(\lambda, St', \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})](\varpi)$ provides $\mathsf{J}_{\mathsf{user}}$ with $\langle i, \mathsf{sec}_i, \mathsf{cert}_i \rangle$.

Correctness. A R-GS scheme is correct if the following conditions are all satisfied:

1. In a valid state St, it always holds that $|St_{users}| = |St_{trans}|$ and two distinct entries of St_{trans} always contain certificates with distinct tag.
2. If the protocol $[\mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(\lambda, St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})]$ is run by two honest parties and $\langle i, \mathsf{cert}_i, \mathsf{sec}_i \rangle$ is obtained by $\mathsf{J}_{\mathsf{user}}$, then it holds that $\mathsf{cert}_i \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_i$.
3. For each revocation epoch t and any $\langle i, \mathsf{cert}_i, \mathsf{sec}_i \rangle$ such that $\mathsf{cert}_i \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_i$, satisfying condition 2, if $i \notin \mathcal{R}_t$, it always holds that $\mathsf{Verify}\big(\mathsf{Sign}(\mathcal{Y}, t, RL_t, \mathsf{cert}_i, \mathsf{sec}_i, M), M, t, RL_t, \mathcal{Y}\big) = 1$.
4. For any outcome $\langle i, \mathsf{cert}_i, \mathsf{sec}_i \rangle$ of the interaction $[\mathsf{J}_{\mathsf{user}}(.,.), \mathsf{J}_{\mathsf{GM}}(., St, ., .)]$ for some valid state St, any revocation epoch t such that $i \notin \mathcal{R}_t$, if $\sigma = \mathsf{Sign}(\mathcal{Y}, t, RL_t, \mathsf{cert}_i, \mathsf{sec}_i, M)$, then

$$\mathsf{Open}(M, t, RL_t, \sigma, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, St') = i.$$

Security Model. As in [40], we formalize security properties via experiments where the adversary interacts with a stateful interface $\mathcal{I}$ that maintains the following variables:

- $\mathsf{state}_{\mathcal{I}}$: is a data structure representing the state of the interface as the adversary invokes the various oracles. It is initialized as $\mathsf{state}_{\mathcal{I}} = (St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(\lambda, N)$. It includes the (initially empty) set St_{users} of group members and a dynamically growing database St_{trans} storing the transcripts of previously executed join protocols. Finally, $\mathsf{state}_{\mathcal{I}}$ includes a counter t (which is initialized to 0) indicating the number of user revocation queries so far.
- $n = |St_{users}| < N$ denotes the current cardinality of the group.
- Sigs: is a database of signatures created by the signing oracle. Each entry consists of a triple (i, t, M, σ) indicating that message M was signed by user i at epoch t.

- U^a: is the set of users that were introduced by the adversary in the system in an execution of the join protocol.
- U^b: is the set of honest users that the adversary, acting as a dishonest group manager, introduced in the system. For these users, the adversary obtains the transcript of the join protocol but not the user's membership secret.

When mounting attacks, adversaries will be granted access to the following oracles.

- Q_{pub}, Q_{keyGM} and Q_{keyOA}: when these oracles are invoked, the interface looks up $\mathsf{state}_{\mathcal{I}}$ and returns the group public key $\mathcal{Y}$, the GM's private key $\mathcal{S}_{\mathsf{GM}}$ and the opening authority's private key $\mathcal{S}_{\mathsf{OA}}$ respectively.
- $Q_{\mathsf{a\text{-}join}}$: allows the adversary to introduce users under his control in the group. On behalf of the GM, the interface runs J_{GM} in interaction with the $\mathsf{J}_{\mathsf{user}}$-executing adversary who plays the role of the prospective user in the join protocol. If this protocol successfully ends, the interface increments N, updates St by inserting the new user n in both sets St_{users} and U^a. It also sets $St_{\mathsf{trans}} := St_{\mathsf{trans}}||\langle n, \mathsf{transcript}_n\rangle$.
- $Q_{\mathsf{b\text{-}join}}$: allows the adversary, acting as a corrupted group manager, to introduce new honest group members of his choice. The interface triggers an execution of $[\mathsf{J}_{\mathsf{user}}, \mathsf{J}_{\mathsf{GM}}]$ and runs $\mathsf{J}_{\mathsf{user}}$ in interaction with the adversary who runs J_{GM}. If the protocol successfully completes, the interface increments n, adds user n to St_{users} and U^b and sets $St_{\mathsf{trans}} := St_{\mathsf{trans}}||\langle n, \mathsf{transcript}_n\rangle$. It stores the membership certificate cert_n and the membership secret sec_n in a *private* part of $\mathsf{state}_{\mathcal{I}}$.
- Q_{sig}: given a message M, an index i, the interface checks if the private area of $\mathsf{state}_{\mathcal{I}}$ contains a certificate cert_i and a membership secret sec_i such that $i \notin \mathcal{R}_t$, where t is the current revocation epoch. If no such elements $(\mathsf{cert}_i, \mathsf{sec}_i)$ exist or if $i \notin U^b$, the interface returns $\perp$. Otherwise, it outputs a signature σ on behalf of user i for epoch t and also sets $\mathsf{Sigs} \leftarrow \mathsf{Sigs}||(i, t, M, \sigma)$.
- Q_{open}: when this oracle is invoked on input of a valid pair (M, σ) for some revocation epoch t, the interface runs algorithm Open using the current state St. When S is a set of triples of the form (M, σ, t), $Q_{\mathsf{open}}^{\neg S}$ denotes a restricted oracle that only applies the opening algorithm to triples (M, σ, t) which are not in S.
- Q_{read} and Q_{write}: are used by the adversary to read and write the content of $\mathsf{state}_{\mathcal{I}}$. Namely, at each invocation, Q_{read} outputs the whole $\mathsf{state}_{\mathcal{I}}$ but the public/private keys and the private part of $\mathsf{state}_{\mathcal{I}}$ where membership secrets are stored after $Q_{\mathsf{b\text{-}join}}$-queries. By using Q_{write}, the adversary can modify $\mathsf{state}_{\mathcal{I}}$ at will as long as it does not remove or alter elements of St_{users}, St_{trans} or invalidate the public state St: for example, the adversary is allowed to create dummy users as long as it does not re-use already existing certificate tags.
- Q_{revoke}: is a revocation oracle. Given an index i such that $i \in St_{users}$, the interface checks if i appears in the appropriate user set (namely, U^a or U^b depending on the considered security notion) and if the database St_{trans} contains a record $\langle i, \mathsf{transcript}_i\rangle$ such that $i \notin \mathcal{R}_t$, where t is the current revocation epoch. If not, it returns $\perp$. Otherwise, it increments t, adds i to $\mathcal{R}_t$ and generates an updated revocation list RL_t which is made available to the adversary. For simplicity, we assumed that the adversary only revokes one user per query to Q_{revoke} but the model easily extends to allow multiple revocations at once.

The Kiayias-Yung model considers properties called security against *misidentification attacks*, *framing attacks* and *anonymity.*

In a misidentification attack, the adversary can corrupt the opening authority using the Q_{keyOA}

oracle. Moreover, he can also introduce malicious users in the group via $Q_{\mathsf{a\text{-}join}}$-queries and revoke users at any time using Q_{revoke}. His purpose is to come up with a signature $\sigma^\star$ that verifies w.r.t. $RL_{t^\star}$, where $t^\star$ denotes the current revocation epoch (*i.e.*, the number of Q_{revoke}-queries). He is deemed successful if the produced signature $\sigma^\star$ does not open to any unrevoked adversarially-controlled.

Definition 6. *A R-GS scheme is secure against misidentification attacks if, for any PPT adversary $\mathcal{A}$ involved in the experiment hereafter, we have* $\mathbf{Adv}_{\mathcal{A}}^{\text{mis-id}}(\lambda) = \Pr[\mathbf{Expt}_{\mathcal{A}}^{\text{mis-id}}(\lambda) = 1] \in \mathsf{negl}(\lambda)$.

```
Experiment $\mathbf{Expt}_{\mathcal{A}}^{\text{mis-id}}(\lambda)$
    $\mathsf{state}_{\mathcal{I}} = (St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(\lambda, N)$;
    $(M^\star, \sigma^\star) \leftarrow \mathcal{A}(Q_{\mathsf{pub}}, Q_{\mathsf{a\text{-}join}}, Q_{\mathsf{revoke}}, Q_{\mathsf{read}}, Q_{\mathsf{keyOA}})$;
    If $\mathsf{Verify}(\sigma^\star, M, t^\star, RL_{t^\star}, \mathcal{Y}) = 0$ return $0$;
    $i = \mathsf{Open}(M^\star, t^\star, RL_{t^\star}, \sigma^\star, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, St')$;
    If $(i \notin U^a \backslash \mathcal{R}_{t^\star})$ return $1$;
    Return $0$;
```

This definition extends the usual definition [40] in that $\mathcal{A}$ also wins if his forgery $\sigma^\star$ verifies w.r.t. $RL_{t^\star}$ but opens to an adversarially-controlled user that *was* revoked during the revocation epoch $t^\star$.

Framing attacks consider the situation where the entire system, including the group manager and the opening authority, is colluding against some honest user. The adversary can corrupt the group manager as well as the opening authority (via oracles Q_{keyGM} and Q_{keyOA}, respectively). He is also allowed to introduce honest group members (via $Q_{\mathsf{b\text{-}join}}$-queries), observe the system while these users sign messages and create dummy users using Q_{write}. In addition, before the possible corruption of the group manager, the adversary can revoke group members at any time by invoking the Q_{revoke} oracle. As a potentially corrupted group manager, $\mathcal{A}$ is allowed to come up with his own revocation list $RL_{t^\star}$ at the end of the game. We assume that anyone can publicly verify that $RL_{t^\star}$ is correctly formed (*i.e.*, that it could be a legitimate output of Revoke) so that the adversary does not come up with an ill-formed revocation list. For consistency, if $\mathcal{A}$ chooses not to corrupt the GM, the produced revocation list $RL_{t^\star}$ must be the one determined by the history of Q_{revoke}-queries. The adversary eventually aims at framing an honest group member.

Definition 7. *A R-GS scheme is secure against framing attacks if, for any PPT adversary $\mathcal{A}$, it holds that* $\mathbf{Adv}_{\mathcal{A}}^{\text{fra}}(\lambda) = \Pr[\mathbf{Expt}_{\mathcal{A}}^{\text{fra}}(\lambda) = 1] \in \mathsf{negl}(\lambda)$.

```
Experiment $\mathbf{Expt}_{\mathcal{A}}^{\text{fra}}(\lambda)$
    $\mathsf{state}_{\mathcal{I}} = (St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(\lambda, N)$;
    $(M^\star, \sigma^\star, t^\star, RL_{t^\star}) \leftarrow \mathcal{A}(Q_{\mathsf{pub}}, Q_{\mathsf{keyGM}}, Q_{\mathsf{keyOA}}, Q_{\mathsf{b\text{-}join}}, Q_{\mathsf{revoke}}, Q_{\mathsf{sig}}, Q_{\mathsf{read}}, Q_{\mathsf{write}})$;
    If $\mathsf{Verify}(\sigma^\star, M^\star, t^\star, RL_{t^\star}, \mathcal{Y}) = 0$ then return $0$;
    $i = \mathsf{Open}(M^\star, t^\star, RL_{t^\star}, \sigma^\star, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, St')$;
    If $i \notin U^b$ return $0$;
    If $\big(\bigwedge_{j \in U^b \text{ s.t. } j=i} (j, t^\star, M^\star, *) \notin \mathsf{Sigs}\big)$ then return $1$;
    Return $0$;
```

The notion of anonymity is formalized by means of a game involving a two-stage adversary. In the following, we assume that, from a given valid membership certificate/secret pair $(\mathsf{cert}, \mathsf{sec})$ and a given revocation list RL_t, it is easy to decide if $(\mathsf{cert}, \mathsf{sec})$ belongs to a revoked user for RL_t. More precisely, there must exist an efficient algorithm $\mathsf{IsRevoked}$ that takes as input $(\mathsf{sec}, \mathsf{cert}, RL_t)$

and returns 1 if the pair $(\mathsf{sec}, \mathsf{cert})$ is not the key material of an unrevoked user for RL_t (such an algorithm obviously exists in our construction).

The first stage of the game is called play stage and allows the adversary $\mathcal{A}$ to modify $\mathsf{state}_{\mathcal{I}}$ via Q_{write}-queries and to open arbitrary signatures by probing Q_{open}. When the play stage ends, the adversary $\mathcal{A}$ chooses a message-period pair $(M^\star, t^\star)$, a revocation list $RL_{t^\star}$ as well as two pairs $(\mathsf{sec}_0^\star, \mathsf{cert}_0^\star)$, $(\mathsf{sec}_1^\star, \mathsf{cert}_1^\star)$, consisting of a valid membership certificate and a corresponding membership secret satisfying $\mathsf{IsRevoked}(\mathsf{sec}_b^\star, \mathsf{cert}_b^\star, RL_{t^\star}) = 0$ for each $b \in \{0,1\}$. Then, the challenger flips a coin $d \xleftarrow{R} \{0,1\}$ and computes a challenge signature $\sigma^\star$ using $(\mathsf{sec}_d^\star, \mathsf{cert}_d^\star)$. The adversary is given $\sigma^\star$ with the task of eventually guessing the bit $d \in \{0,1\}$. Before doing so, he is allowed further oracle queries throughout the second stage, called guess stage, but is restricted not to query Q_{open} for $(M^\star, \sigma^\star, t^\star)$.

Definition 8. *A R-GS scheme is fully anonymous if* $\mathbf{Adv}^{\mathrm{anon}}(\mathcal{A}) := |\Pr[\mathbf{Expt}_{\mathcal{A}}^{\mathrm{anon}}(\lambda) = 1] - 1/2|$ *is negligible for any PPT adversary $\mathcal{A}$ involved in the following experiment:*

```
Experiment $\mathbf{Expt}_{\mathcal{A}}^{\mathrm{anon}}(\lambda)$
    $\mathsf{state}_{\mathcal{I}} = (St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(\lambda)$;
    $\big(aux, M^\star, t^\star, RL_{t^\star}, (\mathsf{sec}_0^\star, \mathsf{cert}_0^\star), (\mathsf{sec}_1^\star, \mathsf{cert}_1^\star)\big)$
        $\leftarrow \mathcal{A}(\mathsf{play} : Q_{\mathsf{pub}}, Q_{\mathsf{keyGM}}, Q_{\mathsf{revoke}}, Q_{\mathsf{open}}, Q_{\mathsf{read}}, Q_{\mathsf{write}})$;
    If $\neg(\mathsf{cert}_b \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_b)$ or $\mathsf{IsRevoked}(\mathsf{sec}_b^\star, \mathsf{cert}_b^\star, RL_{t^\star}) = 1$ for $b \in \{0,1\}$
        or if $\mathsf{cert}_0^\star = \mathsf{cert}_1^\star$ return 0;
    $d \xleftarrow{R} \{0,1\}$;  $\sigma^\star \leftarrow \mathsf{Sign}(\mathcal{Y}, t^\star, \mathsf{cert}_d^\star, \mathsf{sec}_d^\star, M^\star)$;
    $d' \leftarrow \mathcal{A}(\mathsf{guess} : \sigma^\star, aux, Q_{\mathsf{pub}}, Q_{\mathsf{keyGM}}, Q_{\mathsf{open}}^{\neg\{(M^\star, \sigma^\star, t^\star)\}}, Q_{\mathsf{read}}, Q_{\mathsf{write}})$;
    If $d' = d$ then return 1;
    Return 0;
```

B Security Proofs

B.1 Security Against Misidentification Attacks

Theorem 2 (Misidentification). *The scheme is secure against misidentification attacks assuming that the q-SFP and the ℓ-FlexDHE problems are both hard for* $q = \max(q_a, q_r^2)$ *and* $\ell = \log N$*, where q_a and q_r denote the maximal numbers of $Q_{\mathsf{a\text{-}join}}$ queries and Q_{revoke} queries, respectively, and N is the maximal number of group members.*

Proof. Towards a contradiction, let us assume that the adversary $\mathcal{A}$ outputs a non-trivial signature that does not open to an unrevoked adversarially-controlled group member.

Let $\sigma^\star = \big(\mathsf{VK}^\star, \varUpsilon_1^\star, \varUpsilon_2^\star, \varUpsilon_3^\star, \varUpsilon_4^\star, \varUpsilon_5^\star, \varOmega^\star, \mathbf{com}^\star, \mathbf{\Pi}^\star, \sigma_{ots}^\star\big)$ denote $\mathcal{A}$'s forgery and parse $\mathbf{com}^\star$ as

$$\mathbf{com}^\star = \big(com_{C_{v_i}}^\star, com_X^\star, \{com_{R_{l,\tau}}^\star\}_{\tau=2}^5, com_{W_{\phi_l}}^\star, com_{W_{\psi_l}}^\star, com_{\Gamma_l}^\star,$$
$$\{com_{\Psi_{l,\tau}}^\star\}_{\tau \in \{0,1,2\ell\}}, \{com_{\Theta_{l,j}'}^\star\}_{j \in \{1,2,5\}}, \{com_{\theta_{l,j}'}^\star\}_{j \in \{1,2,5\}}, \{com_{\chi_j}^\star\}_{j=1}^3, com_{\sigma_{\mathsf{VK}}}^\star\big)$$

We thus have $\mathsf{Open}(M^\star, t^\star, RL_{t^\star}, \sigma^\star, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, St) \notin U^a \backslash \mathcal{R}_{t^\star}$, where U^a denotes the set of adversarially-controlled users. Depending on the contents of extractable commitments $com_X^\star$, $com_{C_{v_i}}^\star$, $\{com_{R_{l,\tau}}^\star\}_{\tau=2}^5$, $\{com_{\Psi_{l,\tau}}^\star\}_{i \in \{0,1,2\ell\}}$, $com_{W_{\phi_l}}^\star$, $com_{W_{\psi_l}}^\star$, $com_{\Gamma_l}^\star$, we distinguish the following cases:

- **Type I forgeries** are those for which $\{com_{R_{l,\tau}}^\star\}_{\tau=2}^5$ contain group elements $(R_{l,2}^\star, \ldots, R_{l,5}^\star)$ such that $(g^{t^\star}, R_{l,2}^\star, \ldots, R_{l,5}^\star)$ was never signed when the latest revocation list $RL_{t^\star}$ was generated.

- **Type II forgeries** are such that $\{com^\star_{R_{l,\tau}}\}_{\tau=2}^5$ contain group elements $(R^\star_{l,2},\ldots,R^\star_{l,5})$ for which the message $(g^{t^\star},R^\star_{l,2},\ldots,R^\star_{l,5})$ was signed when the latest revocation list $RL_{t^\star}$ was publicized at epoch $t^\star$. At the same time, Open uncovers a user's tag $X^\star$ for which one of the following two situations occurs:
  a. The pair $(X^\star,C^\star_{v_i})$ was not signed using $sk^{(0)}_{\mathsf{AHO}}$.
  b. $(X^\star,C^\star_{v_i})$ was signed when answering some $Q_{\mathsf{a\text{-}join}}$-query. However, $C^\star_{v_i}$ encodes the path $(I^\star_1,\ldots,I^\star_\ell)$ of a leaf $v^\star_i$ assigned to a revoked user $i^\star$ even though the forgery $\sigma^\star$ provides convincing evidence that the committed values $C^\star_{v_i}$, $(R^\star_{l,2},R^\star_{l,3},R^\star_{l,4},R^\star_{l,5})$, $(\Psi^\star_{l,0},\Psi^\star_{l,1},\Psi^\star_{l,2\ell})$ and $(\Gamma^\star_l,W^\star_{\phi_l},W^\star_{\psi_l})$ satisfy the relations

$$e(R^\star_{l,2},C^\star_{v_i}) = e(R^\star_{l,3},g_\ell)\cdot e(g,W^\star_{\phi_l}), \tag{11}$$

and

$$\begin{aligned} e(R^\star_{l,4},C^\star_{v_i}) &= e(\Psi^\star_{l,1},g_\ell)\cdot e(g,W^\star_{\psi_l}) & (12)\\ e(\Psi^\star_{l,0}/R^\star_{l,5},\Gamma^\star_l) &= e(g,g) & (13)\\ e(\Psi^\star_{l,1},g) &= e(g_1,\Psi^\star_{l,0}) & (14)\\ e(\Psi^\star_{l,2\ell},g) &= e(g_{2\ell},\Psi^\star_{l,0}). & (15)\end{aligned}$$

It is immediate that Type I and Type II.a forgeries imply a forger against the AHO signature scheme and the proof is omitted.

Lemma 1 demonstrates that a Type II.b forgery necessarily contradicts the ℓ-FlexDHE assumption. This completes the proof since $\sigma^\star$ cannot constitute a successful misidentification attack without being a Type I or a Type II forgery. □

Lemma 1. *The advantage of any Type II.b forger $\mathcal{A}$ is at most*

$$\mathbf{Adv}^{\text{mis-id-II.b}}_{\mathcal{A}}(\lambda) \le \mathbf{Adv}^{\ell\text{-FlexDHE}}(\lambda)$$

where $\ell=\log N$ and N denotes the maximal number of users.

Proof. The reduction $\mathcal{B}$ takes as input a ℓ-FlexDHE instance $(g,g_1,\ldots,g_\ell,g_{\ell+2},\ldots,g_{2\ell})\in\mathbb{G}^{2\ell}$. To generate the group public key $\mathcal{Y}$ it follows exactly the specification of the Setup algorithm with the difference that, instead of computing ck as per step 3 of the algorithm, it defines $ck=(g_1,\ldots,g_\ell,g_{\ell+2},\ldots,g_{2\ell})\in\mathbb{G}^{2\ell-1}$ using its input and gives $\mathcal{Y}:=\big(g,pk^{(0)}_{\mathsf{AHO}},pk^{(1)}_{\mathsf{AHO}},ck,\mathbf{f},\vec{\varphi},(U,V),\Sigma\big)$ to the Type II.b forger $\mathcal{A}$.

Throughout the game, the adversary can adaptively invoke the Q_{pub}, $Q_{\mathsf{a\text{-}join}}$, Q_{revoke}, Q_{read}, and Q_{keyOA} oracles. Since $\mathcal{B}$ knows $\mathcal{S}_{\mathsf{GM}}=(sk^{(0)}_{\mathsf{AHO}},sk^{(1)}_{\mathsf{AHO}})$ and $\mathcal{S}_{\mathsf{OA}}=(\beta_1,\beta_2)$, it can faithfully answer all adversarial queries. The game ends with the adversary outputting a forgery $\sigma^\star$ for which the committed variables $C^\star_{v_i}$, $(R^\star_{l,2},R^\star_{l,3},R^\star_{l,4},R^\star_{l,5})$, $(\Psi^\star_{l,0},\Psi^\star_{l,1},\Psi^\star_{l,2\ell})$ and $(\Gamma^\star_l,W^\star_{\phi_l},W^\star_{\psi_l})$ satisfy relations (11)-(15) although $\sigma^\star$ opens to some user $i^\star\in U^a\cap\mathcal{R}_{t^\star}$.

Note that $(R^\star_{l,1},R^\star_{l,2},R^\star_{l,3},R^\star_{l,4},R^\star_{l,5})$ must be of the form

$$(R^\star_{l,1},R^\star_{l,2},R^\star_{l,3},R^\star_{l,4},R^\star_{l,5}) = \Big(g^{t^\star},\ g_{\phi_l},\ g_1^{\mathsf{ID}(x^\star_{k_l})},\ g_{\psi_l},\ g^{\mathsf{ID}(x^\star_{k_l})}\Big), \tag{16}$$

for some $\phi_l,\psi_l\in\{1,\ldots,\ell\}$ and some $\mathsf{ID}(x^\star_{k_l}),\mathsf{ID}(x^\star_{u_l})\in\{1,\ldots,2N-1\}$ that $\mathcal{B}$ knows for having chosen them itself at the latest Q_{revoke}-query. By hypothesis, $\sigma^\star$ contains a committed pair $(X^\star,C^\star_{v_i})$

that was signed by $\mathcal{B}$ at some $Q_{\mathsf{a\text{-}join}}$-query. Then, $\mathcal{B}$ recalls $(I_1^\star,\ldots,I_\ell^\star)$ such that $C_{v_i}^\star = \prod_{\kappa=1}^{\ell} g_{\ell+1-\kappa}^{I_\kappa^\star}$ from its interaction with $\mathcal{A}$ at that $Q_{\mathsf{a\text{-}join}}$-query. Since $i^\star \in U^a \cap \mathcal{R}_{t^\star}$, it must hold that either:

- $I_{\phi_l}^\star \neq \mathsf{ID}(x_{k_l}^\star)$: In this case, relations (16) and (11) imply that

$$e(g_{\phi_l}, C_{v_i}^\star) = e(g_1, g_\ell)^{\mathsf{ID}(x_{k_l}^\star)} \cdot e(g, W_{\phi_l}^\star) \tag{17}$$

  for values $\phi_l \in \{1,\ldots,\ell\}$ and $\mathsf{ID}(x_{k_l})^\star \in \{1,\ldots,2N-1\}$ that are available to $\mathcal{B}$. Since it also knows $(I_1^\star,\ldots,I_\ell^\star)$ such that $C_{v_i}^\star = \prod_{\kappa=1}^{\ell} g_{\ell+1-\kappa}^{I_\kappa^\star}$, it can compute $W' = \prod_{\kappa=1,\ \kappa\neq\phi_l}^{\ell} g_{\ell+1-\kappa+\phi_l}^{I_\kappa^\star}$ which satisfies

$$e(g_{\phi_l}, C_{v_i}^\star) = e(g_1, g_\ell)^{I_{\phi_l}^\star} \cdot e(g, W'). \tag{18}$$

  By combining (17) and (18), we find that $g_{\ell+1} = \left(W_{\phi_l}^\star / W'\right)^{1/(I_{\phi_l}^\star - \mathsf{ID}(x_{k_l}^\star))}$ is computable by $\mathcal{B}$ and it solves an instance the ℓ-DHE problem (which is not easier than ℓ-FlexDHE).
- $I_{\psi_l}^\star = \mathsf{ID}(x_{u_l}^\star)$: In this situation, if we define $\varrho = \log_{g_1}(\Psi_{l,1}^\star)$, relations (16) and (12)-(15) imply that

$$e(g_{\psi_l}, C_{v_i}^\star) = e(g_1, g_\ell)^{\varrho} \cdot e(g, W_{\psi_l}^\star) \tag{19}$$
$$g^{\varrho - I_{\psi_l}^\star} \neq 1_{\mathbb{G}} \tag{20}$$
$$\Psi_{l,0}^\star = g^{\varrho} \tag{21}$$
$$\Psi_{l,2\ell}^\star = g_{2\ell}^{\varrho} \tag{22}$$

  Also, similarly to the previous case, $\mathcal{B}$ can compute $W' = \prod_{\kappa=1,\ \kappa\neq\psi_l}^{\ell} g_{\ell+1-\kappa+\psi_l}^{I_\kappa^\star}$ such that

$$e(g_{\psi_l}, C_{v_i}^\star) = e(g_1, g_\ell)^{I_{\psi_l}^\star} \cdot e(g, W'). \tag{23}$$

  If we divide (19) by (23), we obtain the equality $e(g_1, g_\ell)^{\varrho - I_{\psi_l}^\star} = e(g, W'/W_{\phi_l}^\star)$, so that $W'/W_{\phi_l}^\star = g_{\ell+1}^{\varrho - I_{\psi_l}^\star}$. The triple

$$\left(\Psi_{l,0}^\star \cdot g^{-I_{\psi_l}^\star},\ W'/W_{\phi_l}^\star,\ \Psi_{l,2\ell}^\star \cdot g_{2\ell}^{-I_{\psi_l}^\star}\right) = \left(g^{\varrho - I_{\psi_l}^\star},\ g_{\ell+1}^{\varrho - I_{\psi_l}^\star},\ g_{2\ell}^{\varrho - I_{\psi_l}^\star}\right)$$

  thus forms a non-trivial solution to the ℓ-FlexDHE problem.

In either case, we observe that $\mathcal{B}$ solves either the given ℓ-FlexDHE instance or the potentially harder ℓ-DHE problem. □

B.2 Security Against Framing Attacks

The security against framing attacks relies on the SDH assumption and the security of the one-time signature.

Theorem 3 (Non-frameability). *The scheme is secure against framing attacks assuming that: (i) the q_b-SDH assumption holds in $\mathbb{G}$, where q_b is the maximal number of $Q_{\mathsf{b\text{-}join}}$-queries; (ii) Σ is a strongly unforgeable one-time signature.*

Proof. As in [35], we consider two kinds of framing attacks that can be possibly mounted by a non-frameability adversary $\mathcal{A}$.

- **Type I attacks**: $\mathcal{A}$ generates a forgery $\sigma^\star = \big(\mathsf{VK}^\star, \Upsilon_1^\star, \Upsilon_2^\star, \Upsilon_3^\star, \Upsilon_4^\star, \Upsilon_5^\star, \Omega^\star, \mathbf{com}^\star, \mathbf{\Pi}^\star, \sigma_{ots}^\star\big)$ for which the one-time verification key $\mathsf{VK}^\star$ was used by some honest group member $i \in U^b$ when answering a Q_{sig}-query.
- **Type II attacks**: $\mathcal{A}$ outputs a forgery $\sigma^\star = \big(\mathsf{VK}^\star, \Upsilon_1^\star, \Upsilon_2^\star, \Upsilon_3^\star, \Upsilon_4^\star, \Upsilon_5^\star, \Omega^\star, \mathbf{com}^\star, \mathbf{\Pi}^\star, \sigma_{ots}^\star\big)$ for which the one-time verification key $\mathsf{VK}^\star$ was never used by Q_{sig} to answer a signing query on behalf of an honest user $i \in U^b$.

Type I attacks clearly defeat the security of the one-time signature. Lemma 2 shows that a Type II forgery would contradict the Strong Diffie-Hellman assumption. □

Lemma 2. *The scheme is secure against framing attacks of Type II if the q_s-SDH problem is hard. More precisely, the advantage of any adversary after q_s Q_{sig}-queries and q_b $Q_{\mathsf{b\text{-}join}}$-queries is at most* $\mathbf{Adv}^{\text{fra-II}}(\lambda) \leq q_b \cdot \mathbf{Adv}^{\text{q}_\text{s}\text{-SDH}}(\lambda)$.

Proof. Let us assume that a PPT adversary $\mathcal{A}$ comes up with a forgery $(M^\star, \sigma^\star)$ that opens to some honest user $i \in U^b$ who did not issue a signature containing the verification key $\mathsf{VK}^\star$. The same proof as in [35] shows that the Strong Diffie-Hellman assumption can be broken.

Given a q-SDH instance $(\tilde{g}, \tilde{g}^a, \ldots, \tilde{g}^{(a^{q_s})}) \in \mathbb{G}^{q_s+1}$, the reduction $\mathcal{B}$ generates a set of q_s one-time signature keys pairs $(\mathsf{SK}_i, \mathsf{VK}_i) \leftarrow \mathcal{G}(\lambda)$ for $i = 1$ to q_s. Then, using the Boneh-Boyen techniques (see [12][Lemma 3.2]) it builds a generator g and a randomly distributed public value $X^\dagger = g^a$ – which implicitly defines $x^\dagger = \log_g(X^\dagger) = a$ – such that it knows $\{(g^{1/(a+\mathsf{VK}_i)}, \mathsf{VK}_i)\}_{i=1}^{q_s}$.

Next, using the newly generated g, $\mathcal{B}$ generates key pairs $\{(sk_{\mathsf{AHO}}^{(b)}, pk_{\mathsf{AHO}}^{(b)})\}_{b=0,1}$ for the AHO signature (note that group elements of $\{pk_{\mathsf{AHO}}^{(b)}\}_{b=0,1}$ are computed as powers of g) and uses $pk_{\mathsf{AHO}}^{(0)}, pk_{\mathsf{AHO}}^{(1)}$ to form the group public key

$$\mathcal{Y} := \Big(g,\ pk_{\mathsf{AHO}}^{(0)},\ pk_{\mathsf{AHO}}^{(1)},\ ck,\ \mathbf{f},\ \vec{\varphi},\ (U, V),\ \Sigma\Big).$$

The underlying Groth-Sahai CRS $\mathbf{f} = (\vec{f_1}, \vec{f_2}, \vec{f_3})$ is generated for the perfect soundness setting, *i.e.*, with $\vec{f_1} = (f_1 = g^{\beta_1}, 1, g)$, $\vec{f_2} = (1, f_2 = g^{\beta_2}, g)$ and $\vec{f_3} = \vec{f_1}^{\xi_1} \cdot \vec{f_2}^{\xi_2}$, where $\xi_1, \xi_2 \xleftarrow{R} \mathbb{Z}_p^*$.

If the adversary $\mathcal{A}$ decides to corrupt the group manager or the opening authority during the game, $\mathcal{B}$ can reveal $\mathcal{S}_{\mathsf{GM}} = (sk_{\mathsf{AHO}}^{(0)}, sk_{\mathsf{AHO}}^{(1)})$ and $\mathcal{S}_{\mathsf{OA}} = (\beta_1, \beta_2) = (\log_g(f_1), \log_g(f_2))$. At the outset of the game, $\mathcal{B}$ picks a random $j^\star \xleftarrow{R} \{1, \ldots, q_b\}$ and interacts with the Type II forger $\mathcal{A}$ as follows.

- Q_{keyGM}-queries: if $\mathcal{A}$ decides to corrupt the group manager, $\mathcal{B}$ surrenders $\mathcal{S}_{\mathsf{GM}} = (sk_{\mathsf{AHO}}^{(0)}, sk_{\mathsf{AHO}}^{(1)})$.
- $Q_{\mathsf{b\text{-}join}}$-queries: when $\mathcal{A}$, acting as a corrupted group manager, decides to introduce a new honest user i in the group, $\mathcal{B}$ starts interacting with $\mathcal{A}$ in an execution of Join and runs $\mathsf{J_{user}}$ on behalf of the honest user. The actions taken by $\mathcal{B}$ depend on the index $j \in \{1, \ldots, q_b\}$ of the $Q_{\mathsf{b\text{-}join}}$-query.
  - If $j \neq j^\star$, $\mathcal{B}$ follows exactly the specification of $\mathsf{J_{user}}$.
  - If $j = j^\star$, $\mathcal{B}$ sends the value $X^\dagger$ to $\mathsf{J_{GM}}$ at step 1 of Join. User $j^\star$'s membership secret is implicitly defined to be the unknown exponent $\mathsf{sec}_{j^\star} = a$ of the q-SDH instance. In steps 2-5 of the join protocol, $\mathcal{B}$ proceeds like the real $\mathsf{J_{user}}$ algorithm . When Join terminates, $\mathcal{B}$ obtains a membership certificate $\mathsf{cert}_{j^\star} = \big(\mathsf{ID}(v^\star), X^\dagger, C_{v^\star}, \sigma_{v^\star}\big)$.

- Q_{pub}-queries: can be treated as in the real game, by having the simulator return $\mathcal{Y}$.
- Q_{sig}-queries: when the adversary $\mathcal{A}$ asks user $i \in U^b$ to sign a message M, $\mathcal{B}$ can answer the query by running the real signature generation algorithm if $i \neq j^\star$. Otherwise (namely, if $i = j^\star$), $\mathcal{B}$ uses the next available pair $\{(g^{1/(a+\mathsf{VK}_i)}, \mathsf{VK}_i)\}_{i=1}^{q_s}$ to define $\sigma_{\mathsf{VK}_i} = g^{1/(a+\mathsf{VK}_i)}$. It also recalls user $j^\star$'s membership certificate $\mathsf{cert}_{j^\star} = \big(\mathsf{ID}(v^\star), X^\dagger, C_{v^\star}, \sigma_{v^\star}\big)$ that it obtained from the J_{GM}-executing adversary at the $j^\star$-th $Q_{\mathsf{b\text{-}join}}$-query. Using σ_{VK_i} and $\mathsf{cert}_{j^\star}$, it can easily generate all signature components and sign them using the one-time private key SK_i.

Finally, $\mathcal{A}$ outputs a signature $\sigma^\star = \big(\mathsf{VK}^\star, \varUpsilon_1^\star, \varUpsilon_2^\star, \varUpsilon_3^\star, \varUpsilon_4^\star, \varUpsilon_5^\star, \varOmega^\star, \mathbf{com}^\star, \mathbf{\Pi}^\star, \sigma_{ots}^\star\big)$, for some message $M^\star$, that opens to some user $i^\star \in U^b$ who did not sign $M^\star$. At this point, $\mathcal{B}$ halts and reports failure if it turns out that $X^\dagger \neq \varUpsilon_3^\star \cdot \varUpsilon_1^{\star -1/\beta_1} \cdot \varUpsilon_2^{\star -1/\beta_2}$ since, in this case, it was unfortunate when drawing the random index $j^\star$. Still, with probability $1/q_b$, the signature $\sigma^\star$ opens to the user introduced at the $j^\star$-th $Q_{\mathsf{b\text{-}join}}$-query and $(\varUpsilon_1^\star, \varUpsilon_2^\star, \varUpsilon_3^\star)$ does decrypt to $X^\star$. In this situation, the perfect soundness of the proof system ensures that $com_{\sigma_{\mathsf{VK}^\star}}^\star$ is a commitment to a group element $\sigma_{\mathsf{VK}^\star}^\star$ such that $e(\sigma_{\mathsf{VK}^\star}^\star, X^\dagger \cdot g^{\mathsf{VK}^\star}) = e(g, g)$. Since $\sigma^\star$ is a Type II forgery, $\mathcal{B}$ can use β_1, β_2 to compute a BBS decryption of $com_{\sigma_{\mathsf{VK}^\star}}^\star$ and obtain a pair of the form $(\sigma_{\mathsf{VK}^\star}, \mathsf{VK}^\star) = (g^{1/(x+\mathsf{VK}^\star)}, \mathsf{VK}^\star)$. The latter eventually yields a solution $(\tilde{g}^{1/(x+\mathsf{VK}^\star)}, \mathsf{VK}^\star)$ to the initial q_s-SDH instance by performing an Euclidean division in the exponent as in [12]. □

B.3 Anonymity

As for the anonymity property, it naturally relies on the DLIN assumption. The proof is essentially identical to that of Lemma 5 in [35] but we give it for completeness.

Theorem 4 (Anonymity). *The advantage of any anonymity adversary is at most*

$$\mathbf{Adv}^{\mathrm{anon}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{ots}}(\lambda) + 3 \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\lambda),$$

where the first term is $\mathcal{A}$'s probability of breaking the strong unforgeability of the one-time signature.

Proof. We consider a sequence of games at the end of which even an unbounded adversary has no advantage. In Game i, we call S_i the event that $\mathcal{A}$ wins and define $Adv_i = |\Pr[S_i] - 1/2|$.

Game 1: is the experiment of definition 8. In the play stage, the adversary $\mathcal{A}$ can obtain the group public key $\mathcal{Y}$, the group manager's private key $\mathcal{S}_{\mathsf{GM}} = (sk_{\mathsf{AHO}}^{(0)}, sk_{\mathsf{AHO}}^{(1)})$. It can also ask for the opening of any group signature and read/write the content of $\mathsf{state}_{\mathcal{I}}$. When it decides to enter the challenge phase, it outputs a message $M^\star$, a period index $t^\star$ and two membership certificate/secret $(\mathsf{cert}_0^\star, \mathsf{sec}_0^\star)$ and $(\mathsf{cert}_1^\star, \mathsf{sec}_1^\star)$ such that $\mathsf{cert}_b^\star \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_b^\star$ for $b = 0, 1$. The simulator $\mathcal{B}$ flips a fair coin $d \stackrel{R}{\leftarrow} \{0,1\}$ and computes $\sigma^\star \leftarrow \mathsf{Sign}(\mathcal{Y}, t^\star, RL_{t^\star}, \mathsf{cert}_d^\star, \mathsf{sec}_d^\star, M^\star)$, where $t^\star$ is determined by the history of Q_{revoke}-queries. The signature $\sigma^\star$ is given as a challenge to $\mathcal{A}$ who has to guess $d \in \{0,1\}$ after another series of queries (under the natural restriction of not querying the opening of $\sigma^\star$). We have $Adv_1 = \mathbf{Adv}^{\mathrm{anon}}(\mathcal{A})$.

Game 2: is as **Game** 1 but $\mathcal{B}$ halts if $\mathcal{A}$ queries the opening of a signature σ containing the same one-time verification key $\mathsf{VK}^\star$ as in the challenge phase (we assume w.l.o.g. that $(\mathsf{SK}^\star, \mathsf{VK}^\star)$ is generated at the outset of the game). If such a query is made before the challenge phase, it means that $\mathcal{A}$ was able to forge a one-time signature even without having seen a signature. If the query occurs after the challenge phase, then the strong unforgeability of $\varSigma$ is broken. We can thus write

$|\Pr[S_2] - \Pr[S_1]| \leq \mathbf{Adv}^{\mathrm{ots}}(\lambda)$.

Game 3: we change the generation of $\mathcal{Y}$ so as to answer Q_{open}-queries without using the secret exponents $\beta_1, \beta_2 \in \mathbb{Z}_p$ that define $\mathcal{S}_{\mathsf{OA}}$. To this end, $\mathcal{B}$ chooses $\alpha_u, \alpha_v \overset{R}{\leftarrow} \mathbb{Z}_p^*$, and defines $U = g^{-\mathsf{VK}^\star} \cdot f_1^{\alpha_u}$, and $V = g^{-\mathsf{VK}^\star} \cdot f_2^{\alpha_v}$. It is not hard to see (see [42] for details) that, for any Q_{open}-query containing a BBS encryption $(\Upsilon_1, \Upsilon_2, \Upsilon_3) = (f_1^{z_1}, f_2^{z_2}, X \cdot g^{z_1+z_2})$, the values (Υ_4, Υ_5) reveal g^{z_1} and g^{z_2} (and thus the encrypted X) since $\mathsf{VK} \neq \mathsf{VK}^\star$ unless the event introduced in Game 2 occurs. To generate the challenge signature $\sigma^\star$ at epoch $t^\star$, the challenger $\mathcal{B}$ first computes $(\Upsilon_1^\star, \Upsilon_2^\star, \Upsilon_3^\star)$ and then $(\Upsilon_4^\star, \Upsilon_5^\star) = (\Upsilon_1^{\star\alpha_u}, \Upsilon_2^{\star\alpha_v})$. It sets the challenge signature to be $\sigma^\star = (\mathsf{VK}^\star, \Upsilon_1^\star, \Upsilon_2^\star, \Upsilon_3^\star, \Upsilon_4^\star, \Upsilon_5^\star, \Omega^\star, \mathbf{com}^\star, \mathbf{\Pi}^\star, \sigma_{ots}^\star)$. It can be checked that the distributions of $\mathcal{Y}$ and $\sigma^\star$ are unchanged and we have $\Pr[S_3] = \Pr[S_2]$.

Game 4: in the setup phase, we generate the CRS $\mathbf{f} = (\vec{f_1}, \vec{f_2}, \vec{f_3})$ of the proof system for the perfect WI setting. We choose $\vec{f_3} = \vec{f_1}^{\xi_1} \cdot \vec{f_2}^{\xi_2} \cdot (1,1,g)^{-1}$ instead of $\vec{f_3} = \vec{f_1}^{\xi_1} \cdot \vec{f_2}^{\xi_2}$ so that $\vec{f_1}$, $\vec{f_2}$ and $\vec{f_3}$ are linearly independent. Any significant change in $\mathcal{A}$'s behavior yields a distinguisher for the DLIN problem and we can write $|\Pr[S_4] - \Pr[S_3]| = 2 \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\mathcal{B})$. As noted in [36], proofs in the WI setting reveal no information on which witnesses they were generated from.

Game 5: in this game, we modify the generation of the challenge signature $\sigma^\star$ and use the trapdoor of the Groth-Sahai CRS (namely, the exponents ξ_1, ξ_2 for which $\vec{\varphi} = \vec{f_1}^{\xi_1} \cdot \vec{f_2}^{\xi_2}$) to generate simulated proofs $\{\pi_{eq\text{-}com,j}\}_{j=1}^3$ that $(\Upsilon_1^\star, \Upsilon_2^\star, \Upsilon_3^\star)$ and com_X encrypt of the same value. It is known [36] that linear multi-exponentiation equations always have perfectly NIZK proofs on a simulated CRS. For, any satisfiable relation, (ξ_1, ξ_2) allows generating proofs without using the witnesses χ_1, χ_2, χ_3 for which (10) holds and simulated proofs are perfectly indistinguishable from real ones. Hence, $\Pr[S_5] = \Pr[S_4]$.

Game 6: in the computation of $\Upsilon_3^\star$, we now replace $g^{z_1+z_2}$ by a random group element in the challenge $\sigma^\star$. Since $\mathcal{B}$ does not explicitly use $z_1 = \log_{f_1}(\Upsilon_1^\star)$, $z_2 = \log_{f_2}(\Upsilon_2^\star)$, any change in $\mathcal{A}$'s behavior yields a distinguisher for the DLIN problem and $|\Pr[S_6] - \Pr[S_5]| \leq \mathbf{Adv}^{\mathrm{DLIN}}(\mathcal{B})$. In Game 6, we have $\Pr[S_6] = 1/2$. Indeed, when we consider the challenge $\sigma^\star$, Groth-Sahai commitments are all perfectly hiding in the WI setting and proofs $\mathbf{\Pi}$ reveal nothing about the underlying witnesses (in particular, NIZK proofs $\{\pi_{eq\text{-}com,j}\}_{j=1}^3$ are generated without using them) and $(\Upsilon_1^\star, \Upsilon_2^\star, \Upsilon_3^\star)$ perfectly hides $X^\star$. Finally, randomized signature components $\Omega^\star = \{{\Theta'_{l,i}}^\star, {\theta'_{l,i}}^\star\}_{i \in \{3,4,6,7\}}$ are information-theoretically independent of the corresponding messages and the remaining components of AHO signatures $\Theta_l^\star$ and $\theta_l^\star$.

When combining the above, $\mathcal{A}$'s advantage can be bounded by $\mathbf{Adv}^{\mathrm{anon}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{ots}}(\lambda) + 3 \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\lambda)$ as stated by the theorem. □

C Constructions from Weaker Assumptions

C.1 CDH-Based Vector Commitments

In [27], Catalano and Fiore described a vector commitment scheme whose binding property relies on the Diffie-Hellman assumption. In their scheme, if ℓ is the dimension of committed vectors, a commitment key

$$\left(g, g_1, \ldots, g_\ell, h_1, \ldots, h_\ell, \{h_{i,j}\}_{i \neq j}^{\ell}\right) \in \mathbb{G}^{1+\ell+\ell^2}$$

is obtained by randomly choosing $\alpha_1, \ldots, \alpha_\ell \overset{R}{\leftarrow} \mathbb{Z}_p^*$ and defining $g_i = g^{\alpha_i}$, $h_i = g^{\prod_{\kappa \neq i} \alpha_\kappa}$ and $h_{i,j} = g^{\prod_{\kappa \neq i,j}^{\ell} \alpha_\kappa} = h_j^{1/\alpha_i}$ (so that $h_{i,j} = h_{i,j}$) for each $i \in \{1, \ldots, \ell\}$ and $j \neq i$. A commitment to

$\vec{m} = (m_1, \ldots, m_\ell)$ is obtained as $C = \prod_{\kappa=1}^{\ell} g_\kappa^{m_\kappa}$. By revealing $W_i = \prod_{\kappa=1,\ \kappa \neq i}^{\ell} h_{i,\kappa}^{m_\kappa}$, the committer can open the commitment to m_i at the i-th coordinate of $\vec{m}$ as it satisfies the equation

$$e(g, C) \cdot e(g^{-m_i}, h_i) = e(g_i, W_i).$$

This time, the coordinate-wise binding property relies on the standard Computational Diffie-Hellman (CDH) assumption. Note that, in its basic version, the commitment is not (and does not need to be) hiding since it does not use any randomizer.

C.2 Construction

This section gives an alternative construction of revocable group signature where the ℓ-FlexDHE assumption is not used. Instead, we rely on an assumption (suggested in [44]) of fixed size, which is inspired by the Flexible Diffie-Hellman assumption [43].

Definition 9 ([44]). *In a group $\mathbb{G}$ of prime order p, the Flexible Square Diffie-Hellman (FSDH) problem consists in, given (g, g^a) with $a \stackrel{R}{\leftarrow} \mathbb{Z}_p$, finding a non-trivial triple $(g^\mu, g^{a\cdot\mu}, g^{(a^2)\cdot\mu})$, with $\mu \neq 0$.*

The Flexible Square Diffie-Hellman assumption is the hardness of FSDH for any PPT algorithm.

We thus trade one of the q-type assumptions for a constant-size assumption at the cost of increasing the size of the group public key. Indeed, the latter now contains $O(\log^2 N)$ group elements.

Setup(λ, N)**:** given a security parameter $\lambda \in \mathbb{N}$ and the maximal number of users $N = 2^{\ell-1}$,

1. Choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$, with a generator $g \stackrel{R}{\leftarrow} \mathbb{G}$.
2. Define $n_0 = 2$ and $n_1 = 7$. Generate two key pairs $(sk_{\mathsf{AHO}}^{(0)}, pk_{\mathsf{AHO}}^{(0)})$ and $(sk_{\mathsf{AHO}}^{(1)}, pk_{\mathsf{AHO}}^{(1)})$ for the AHO signature in order to sign messages of n_0 and n_1 group elements, respectively. These key pairs are
$$pk_{\mathsf{AHO}}^{(d)} = \Big(G_r^{(d)},\ H_r^{(d)},\ G_z^{(d)} = G_r^{\gamma_z^{(d)}},\ H_z^{(d)} = H_r^{\delta_z^{(d)}},$$
$$\{G_i^{(d)} = G_r^{\gamma_i^{(d)}}, H_i^{(d)} = H_r^{\delta_i^{(d)}}\}_{i=1}^{n_d},\ A^{(d)},\ B^{(d)}\Big)$$
and $sk_{\mathsf{AHO}}^{(d)} = \big(\alpha_a^{(d)}, \alpha_b^{(d)}, \gamma_z^{(d)}, \delta_z^{(d)}, \{\gamma_i^{(d)}, \delta_i^{(d)}\}_{i=1}^{n_d}\big)$, where $d \in \{0,1\}$. These two schemes will be used to sign messages consisting of 2 and 7 group elements, respectively.
3. Generate a public key $ck = \big(g_1, \ldots, g_\ell, h_1, \ldots, h_\ell, \{h_{i,j}\}_{i \neq j}^{\ell}\big) \in \mathbb{G}^{\ell+\ell^2}$ for vectors of dimension ℓ in the CDH-based vector commitment scheme recalled in Section C.2.
4. As a CRS for the NIWI proof system, select vectors $\mathbf{f} = (\vec{f_1}, \vec{f_2}, \vec{f_3})$ s.t. $\vec{f_1} = (f_1, 1, g) \in \mathbb{G}^3$, $\vec{f_2} = (1, f_2, g) \in \mathbb{G}^3$, and $\vec{f_3} = \vec{f_1}^{\xi_1} \cdot \vec{f_2}^{\xi_2}$, with $f_1 = g^{\beta_1}, f_2 = g^{\beta_2} \stackrel{R}{\leftarrow} \mathbb{G}$ and $\beta_1, \beta_2, \xi_1, \xi_2 \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$. We also define the vector $\vec{\varphi} = \vec{f_3} \cdot (1, 1, g)$.
5. Choose $(U, V) \stackrel{R}{\leftarrow} \mathbb{G}^2$ that, together with generators $f_1, f_2, g \in \mathbb{G}$, will form a public encryption key.
6. Select a strongly unforgeable one-time signature $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$.
7. Set $\mathcal{S}_{\mathsf{GM}} := \big(sk_{\mathsf{AHO}}^{(0)}, sk_{\mathsf{AHO}}^{(1)}\big)$, $\mathcal{S}_{\mathsf{OA}} := \big(\beta_1, \beta_2\big)$ as authorities' private keys and the group public key is
$$\mathcal{Y} := \Big(g,\ pk_{\mathsf{AHO}}^{(0)},\ pk_{\mathsf{AHO}}^{(1)},\ ck = \big(g_1, \ldots, g_\ell, h_1, \ldots, h_\ell, \{h_{i,j}\}_{i \neq j}^{\ell}\big),\ \mathbf{f},\ \vec{\varphi},\ (U, V),\ \Sigma\Big).$$

Join$^{(\mathrm{GM},\mathcal{U}_i)}$**:** the group manager and the prospective user $\mathcal{U}_i$ run the following interactive protocol $[\mathsf{J}_{\mathsf{user}}(\lambda,\mathcal{Y}),\mathsf{J}_{\mathsf{GM}}(\lambda,St,\mathcal{Y},\mathcal{S}_{\mathsf{GM}})]$:

1. $\mathsf{J}_{\mathsf{user}}(\lambda,\mathcal{Y})$ picks $x \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and computes $X = g^x$ which is sent to $\mathsf{J}_{\mathsf{GM}}(\lambda,St,\mathcal{Y},\mathcal{S}_{\mathsf{GM}})$. If $X \in \mathbb{G}$ already appears in the database St_{trans}, J_{GM} halts and returns $\perp$ to $\mathsf{J}_{\mathsf{user}}$.
2. J_{GM} assigns to $\mathcal{U}_i$ an available leaf v of identifier $\mathsf{ID}(v)$ in the tree T. Let $x_1,\ldots,x_\ell$ be the path from $x_\ell = v$ to the root $x_1 = \epsilon$ of T. Let also $(I_1,\ldots,I_\ell) = (\mathsf{ID}(x_1),\ldots,\mathsf{ID}(x_\ell))$ be the vector of identifiers (with $I_1 = 1$ and $I_\ell = \mathsf{ID}(v) \in \{N,\ldots,2N-1\}$). Then, J_{GM} conducts the following steps.
   a. Compute a compact encoding of $(I_1,\ldots,I_\ell)$ as $C_v = \prod_{\kappa=1}^{\ell} g_\kappa^{I_\kappa} \in \mathbb{G}$.
   b. Using $sk_{\mathsf{AHO}}^{(0)}$, generate an AHO signature $\sigma_v = (\theta_{v,1},\ldots,\theta_{v,7})$ on $(X,C_v) \in \mathbb{G}^2$ in order to bind C_v to the value X that identifies the new member $\mathcal{U}_i$.
3. J_{GM} sends $\mathsf{ID}(v) \in \{N,\ldots,2N-1\}$ and C_v to $\mathsf{J}_{\mathsf{user}}$ that halts if $\mathsf{ID}(v) \notin \{N,\ldots,2N-1\}$ or if $C_v \neq \prod_{\kappa=1}^{\ell} g_\kappa^{I_\kappa} \in \mathbb{G}$. Otherwise, $\mathsf{J}_{\mathsf{user}}$ sends a signature $sig_i = \mathrm{Sign}_{\mathsf{usk}[i]}\big(X||(I_1,\ldots,I_\ell)\big)$ to J_{GM}.
4. J_{GM} checks that $\mathrm{Verify}_{\mathsf{upk}[i]}\big((X||(I_1,\ldots,I_\ell)),sig_i\big) = 1$. If not J_{GM} aborts. Otherwise, J_{GM} returns σ_v to $\mathsf{J}_{\mathsf{user}}$ and stores $\mathsf{transcript}_i = (X,\mathsf{ID}(v),C_v,\sigma_v,sig_i)$ in the database St_{trans}.
5. $\mathsf{J}_{\mathsf{user}}$ defines the membership certificate as $\mathsf{cert}_i = \big(\mathsf{ID}(v),X,C_v,\sigma_v\big) \in \{N,\ldots,2N-1\}\times\mathbb{G}^9$, where X will serve as the tag identifying $\mathcal{U}_i$. The membership secret sec_i is defined as $\mathsf{sec}_i = x \in \mathbb{Z}_p$.

Revoke$(\mathcal{Y},\mathcal{S}_{\mathsf{GM}},t,\mathcal{R}_t)$**:** Parse $\mathcal{S}_{\mathsf{GM}}$ as $\mathcal{S}_{\mathsf{GM}} := \big(sk_{\mathsf{AHO}}^{(0)},sk_{\mathsf{AHO}}^{(1)}\big)$ and do the following.

1. Find a partition of the unrevoked user set $\{1,\ldots,N\}\backslash\mathcal{R}_t$ as the union of disjoint subsets of the form $S_{k_1,u_1},\ldots,S_{k_m,u_m}$, with $m \leq 2\cdot|\mathcal{R}_t| - 1$.
2. For $i = 1$ to m, do the following.
   a. Parse S_{k_i,u_i} as the difference between sub-trees rooted at an internal node x_{k_i} and one of its descendants x_{u_i}. Let $\phi_i,\psi_i \in \{1,\ldots,\ell\}$ be the depths of x_{k_i} and x_{u_i}, respectively, in T assuming that the root ϵ is at depth 1. Encode S_{k_i,u_i} as a vector of group elements
   $$\big(g_{\phi_i},h_{\phi_i},g^{-\mathsf{ID}(x_{k_i})},g_{\psi_i},h_{\psi_i},g^{\mathsf{ID}(x_{u_i})}\big) \in \mathbb{G}^6.$$
   b. To authenticate S_{k_i,u_i} and link it to the revocation epoch t, use $sk_{\mathsf{AHO}}^{(1)}$ to compute a structure-preserving signature $\Theta_i = (\Theta_{i,1},\ldots,\Theta_{i,7}) \in \mathbb{G}^7$ on the message
   $$R_i = \big(g^t,g_{\phi_i},h_{\phi_i},g^{-\mathsf{ID}(x_{k_i})},g_{\psi_i},h_{\psi_i},g^{\mathsf{ID}(x_{u_i})}\big) \in \mathbb{G}^7,$$
   where the epoch number t is interpreted as an element of $\mathbb{Z}_p$.

Return
$$RL_t = \Big(t,\ \mathcal{R}_t,\ \{\phi_i,\ \psi_i,\ \mathsf{ID}(x_{k_i}),\ \mathsf{ID}(x_{u_i}),\ \Theta_i = (\Theta_{i,1},\ldots,\Theta_{i,7})\}_{i=1}^m\Big). \tag{24}$$

Sign$(\mathcal{Y},t,RL_t,\mathsf{cert}_i,\mathsf{sec}_i,M)$**:** return $\perp$ if $i \in \mathcal{R}_t$. Otherwise, to sign M, generate a one-time key pair $(\mathsf{SK},\mathsf{VK}) \leftarrow \mathcal{G}(\lambda)$. Parse cert_i as $\mathsf{cert}_i = \big(\mathsf{ID}(v_i),X,C_{v_i},\sigma_{v_i}\big) \in \{N,\ldots,2N-1\}\times\mathbb{G}^9$ and sec_i as $x \in \mathbb{Z}_p$. Let $\epsilon = x_1,\ldots,x_\ell = v_i$ be the path connecting the leaf v_i to the root ϵ and let $(I_1,\ldots,I_\ell) = (\mathsf{ID}(x_1),\ldots,\mathsf{ID}(x_\ell))$. First, $\mathcal{U}_i$ generates a commitment $com_{C_{v_i}}$ to the encoding C_{v_i} of the path $(I_1,\ldots,I_\ell)$ from v_i to the root. Then, conduct the following steps.

1. Using RL_t, find the set S_{k_l,u_l}, with $l \in \{1,\ldots,m\}$, that contains the leaf v_i identified by v_i. Let x_{k_l} and x_{u_l} denote the primary and secondary roots of S_{k_l,u_l} at depths ϕ_l and ψ_l, respectively. Since x_{k_l} is an ancestor of v_i but x_{u_l} is not, it holds that $I_{\phi_l} = \mathsf{ID}(x_{k_l})$ and $I_{\psi_l} \neq \mathsf{ID}(x_{u_l})$.
2. To prove that v_i belongs to S_{k_l,u_l}, $\mathcal{U}_i$ first re-randomizes the l-th AHO signature Θ_l of RL_t as $\{\Theta'_{l,i}\}_{i=1}^{7} \leftarrow \mathsf{ReRand}(pk_{\mathsf{AHO}}^{(1)}, \Theta_l)$. Then, he commits to the l-th revocation message

$$R_l = (R_{l,1}, R_{l,2}, R_{l,3}, R_{l,4}, R_{l,5}, R_{l,6}, R_{l,7}) = \big(g^t,\ g_{\phi_l},\ h_{\phi_l},\ g^{-\mathsf{ID}(x_{k_l})},\ g_{\psi_l},\ h_{\psi_l},\ g^{\mathsf{ID}(x_{u_l})}\big) \tag{25}$$

   and its signature $\Theta'_l = (\Theta'_{l,1}, \ldots, \Theta'_{l,7})$ by computing Groth-Sahai commitments $\{com_{R_{l,\tau}}\}_{\tau=2}^{7}$, $\{com_{\Theta'_{l,j}}\}_{j\in\{1,2,5\}}$ to $\{R_{l,\tau}\}_{\tau=2}^{7}$ and $\{\Theta'_{l,j}\}_{j\in\{1,2,5\}}$.

   a. To prove that $I_{\phi_l} = \mathsf{ID}(x_{k_l})$, $\mathcal{U}_i$ first computes $W_{\phi_l} = \prod_{\kappa=1,\ \kappa\neq\phi_l}^{\ell} h_{\phi_l,\kappa}^{I_\kappa}$ that satisfies the equality $e(g, C_{v_i}) \cdot e(g^{-I_{\phi_l}}, h_{\phi_l}) = e(g_{\phi_l}, W_{\phi_l})$. Then, $\mathcal{U}_i$ generates a Groth-Sahai commitment $com_{W_{\phi_l}}$ to W_{ϕ_l}. He computes a proof that committed variables $(R_{l,2}, R_{l,3}, R_{l,4}, C_{v_i}, W_{\phi_l})$ satisfy the equation

$$e(g, C_{v_i}) \cdot e(R_{l,4}, R_{l,3}) = e(R_{l,2}, W_{\phi_l}). \tag{26}$$

   Let π_{eq} be the proof for the quadratic equation (26).

   b. To prove that $I_{\psi_l} \neq \mathsf{ID}(x_{u_l})$, $\mathcal{U}_i$ computes $W_{\psi_l} = \prod_{\kappa=1,\ \kappa\neq\psi_l}^{\ell} h_{\psi_l,\kappa}^{I_\kappa}$ that satisfies the equality $e(g, C_{v_i}) \cdot e(g^{-I_{\psi_l}}, h_{\psi_l}) = e(g_{\psi_l}, W_{\psi_l})$. Then, he computes a commitment $com_{W_{\psi_l}}$ to W_{ψ_l} as well as commitments com_{Γ_l} and $\{com_{\Psi_{l,\tau}}\}_{\tau=0,1}$ to the group elements

$$(\Gamma_l, \Psi_{l,0}, \Psi_{l,1}) = \big(g^{1/(\mathsf{ID}(x_{u_l})-I_{\psi_l})}, g^{-I_{\psi_l}}, g_{\psi_l}^{-I_{\psi_l}}\big).$$

   Then, $\mathcal{U}_i$ provides evidence that committed variables $(R_{l,5}, R_{l,6}, R_{l,7}, C_{v_i}, \Gamma_l, \Psi_l)$ satisfy

$$e(g, C_{v_i}) \cdot e(\Psi_{l,0}, R_{l,6}) = e(R_{l,5}, W_{\phi_l}), \tag{27}$$

$$e(R_{l,7} \cdot \Psi_{l,0}, \Gamma_l) = e(g, g) \tag{28}$$

$$e(\Psi_{l,0}, R_{l,5}) = e(g, \Psi_{l,1}). \tag{29}$$

   We denote this proof by $\pi_{neq} = (\pi_{neq,1}, \pi_{neq,2}, \pi_{neq,3})$. It consists of 27 group elements since all equations are quadratic.

3. $\mathcal{U}_i$ proves that the tuple R_l of (25) is part of RL_t: namely, $\mathcal{U}_i$ computes a proof π_{R_l} that committed message elements $\{R_{l,\tau}\}_{\tau=2}^{7}$ and signature components $\{\Theta'_{l,j}\}_{j\in\{1,2,5\}}$ satisfy the equations

$$A^{(1)} \cdot e(\Theta'_{l,3}, \Theta'_{l,4})^{-1} \cdot e(G_1^{(1)}, g^t)^{-1} = e(G_z^{(1)}, \Theta'_{l,1}) \cdot e(G_r^{(1)}, \Theta'_{l,2}) \cdot \prod_{\tau=2}^{7} e(G_\tau^{(1)}, R_{l,\tau}), \tag{30}$$

$$B^{(1)} \cdot e(\Theta'_{l,6}, \Theta'_{l,7})^{-1} \cdot e(H_1^{(1)}, g^t)^{-1} = e(H_z^{(1)}, \Theta'_{l,1}) \cdot e(H_r^{(1)}, \Theta'_{l,5}) \cdot \prod_{\tau=2}^{7} e(H_\tau^{(1)}, R_{l,\tau}),$$

   The proof π_{R_l} takes 6 elements as both equations of (30) are linear.

4. Let $\sigma_{v_i} = (\theta_{v_i,1}, \ldots, \theta_{v_i,7})$ be the AHO signature on the message (X, C_{v_i}). Set $\{\theta'_{v_i,j}\}_{j=1}^{7} \leftarrow \mathsf{ReRand}(pk_{\mathsf{AHO}}^{(0)}, \sigma_{v_i})$ and generate commitments $\{com_{\theta'_{v_i,j}}\}_{j\in\{1,2,5\}}$ to $\{\theta'_{v_i,j}\}_{j\in\{1,2,5\}}$ as well as a commitment com_X to X. Then, generate a proof $\pi_{\sigma_{v_i}}$ that committed variables satisfy
$$A^{(0)} \cdot e(\theta'_{l,3}, \theta'_{l,4})^{-1} = e(G_z^{(0)}, \theta'_{l,1}) \cdot e(G_r^{(0)}, \theta'_{l,2}) \cdot e(G_1^{(0)}, X) \cdot e(G_2^{(0)}, C_{v_i}),$$
$$B^{(0)} \cdot e(\theta'_{l,6}, \theta'_{l,7})^{-1} = e(H_z^{(0)}, \theta'_{l,1}) \cdot e(H_r^{(0)}, \theta'_{l,5}) \cdot e(H_1^{(0)}, X) \cdot e(H_2^{(0)}, C_{v_i})$$
Since these equations are linear, $\pi_{\sigma_{v_i}}$ requires 6 group elements.
5. Using VK as a tag, compute a tag-based encryption [42] of X by picking $z_1, z_2 \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and setting
$$(\Upsilon_1, \Upsilon_2, \Upsilon_3, \Upsilon_4, \Upsilon_5) = \big(f_1^{z_1}, f_2^{z_2}, X \cdot g^{z_1+z_2}, (g^{\mathsf{VK}} \cdot U)^{z_1}, (g^{\mathsf{VK}} \cdot V)^{z_2}\big).$$
6. Generate a NIZK proof that $com_X = (1, 1, X) \cdot \vec{f_1}^{\,w_{X,1}} \cdot \vec{f_2}^{\,w_{X,2}} \cdot \vec{f_3}^{\,w_{X,3}}$ and $(\Upsilon_1, \Upsilon_2, \Upsilon_3)$ are BBS encryptions of the same value X. If we write $\vec{f_3} = (f_{3,1}, f_{3,2}, f_{3,3})$, the Groth-Sahai commitment com_X can be written as $(f_1^{w_{X,1}} \cdot f_{3,1}^{w_{X,3}}, f_2^{w_{X,2}} \cdot f_{3,2}^{w_{X,3}}, X \cdot g^{w_{X,1}+w_{X,2}} \cdot f_{3,3}^{w_{X,3}})$, so that we have
$$com_X \cdot (\Upsilon_1, \Upsilon_2, \Upsilon_3)^{-1} = \big(f_1^{\chi_1} \cdot f_{3,1}^{\chi_3},\ f_2^{\chi_2} \cdot f_{3,2}^{\chi_3},\ g^{\chi_1+\chi_2} \cdot f_{3,3}^{\chi_3}\big) \tag{31}$$
with $\chi_1 = w_{X,1} - z_1$, $\chi_2 = w_{X,2} - z_2$, $\chi_3 = w_{X,3}$. The signer $\mathcal{U}_i$ commits to $\chi_1, \chi_2, \chi_3 \in \mathbb{Z}_p$ (by computing com_{χ_j}, for $j \in \{1, 2, 3\}$), and generates proofs $\{\pi_{eq\text{-}com,j}\}_{j=1}^{3}$ that χ_1, χ_2, χ_3 satisfy the relations (31).
7. Compute a weak Boneh-Boyen signature $\sigma_{\mathsf{VK}} = g^{1/(x+\mathsf{VK})}$ on VK and a commitment $com_{\sigma_{\mathsf{VK}}}$ to σ_{VK}. Then, generate a NIWI proof $\pi_{\sigma_{\mathsf{VK}}} = (\vec{\pi}_{\sigma_{\mathsf{VK}},1}, \vec{\pi}_{\sigma_{\mathsf{VK}},2}, \vec{\pi}_{\sigma_{\mathsf{VK}},3}) \in \mathbb{G}^9$ that committed variables $(\sigma_{\mathsf{VK}}, X) \in \mathbb{G}^2$ satisfy the quadratic equation $e(\sigma_{\mathsf{VK}}, X \cdot g^{\mathsf{VK}}) = e(g, g)$.
8. Compute $\sigma_{ots} = \mathcal{S}(\mathsf{SK}, (M, RL_t, \Upsilon_1, \Upsilon_2, \Upsilon_3, \Upsilon_4, \Upsilon_5, \Omega, \mathbf{com}, \mathbf{\Pi}))$ where $\Omega = \{\Theta'_{l,i}, \theta'_{l,i}\}_{i\in\{3,4,6,7\}}$ and
$$\mathbf{com} = \big(com_{C_{v_i}}, com_X, \{com_{R_{l,\tau}}\}_{\tau=2}^{7}, com_{W_{\phi_l}}, com_{W_{\psi_l}}, com_{\Gamma_l}, \{com_{\Psi_{l,\tau}}\}_{\tau\in\{0,1\}}, \{com_{\Theta'_{l,j}}\}_{j\in\{1,2,5\}}, \{com_{\theta'_{l,j}}\}_{j\in\{1,2,5\}}, \{com_{\chi_j}\}_{j=1}^{3}, com_{\sigma_{\mathsf{VK}}}\big)$$
$$\mathbf{\Pi} = \big(\pi_{eq}, \pi_{neq}, \pi_{R_l}, \pi_{\sigma_{v_i}}, \{\pi_{eq\text{-}com,j}\}_{j=1}^{3}, \pi_{\sigma_{\mathsf{VK}}}\big)$$

Return the signature $\sigma = \big(\mathsf{VK}, \Upsilon_1, \Upsilon_2, \Upsilon_3, \Upsilon_4, \Upsilon_5, \Omega, \mathbf{com}, \mathbf{\Pi}, \sigma_{ots}\big)$.

Verify$(\sigma, M, t, RL_t, \mathcal{Y})$**:** parse σ as above and do the following.

1. If $\mathcal{V}(\mathsf{VK}, (M, RL_t, \Upsilon_1, \Upsilon_2, \Upsilon_3, \Upsilon_4, \Upsilon_5, \Omega, \mathbf{com}, \mathbf{\Pi}), \sigma_{ots}) = 0$, return 0.
2. Return 0 if $e(\Upsilon_1, g^{\mathsf{VK}} \cdot U) \neq e(f_1, \Upsilon_4)$ or $e(\Upsilon_2, g^{\mathsf{VK}} \cdot V) \neq e(f_2, \Upsilon_5)$.
3. Return 1 if all proofs properly verify. Otherwise, return 0.

Open$(M, t, RL_t, \sigma, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, St)$**:** parse σ as above and return $\perp$ if $\mathsf{Verify}(\sigma, M, t, RL_t, \mathcal{Y}) = 0$. Otherwise, given $\mathcal{S}_{\mathsf{OA}} = (\beta_1, \beta_2)$, compute $\tilde{X} = \Upsilon_3 \cdot \Upsilon_1^{-1/\beta_1} \cdot \Upsilon_2^{-1/\beta_2}$. In the database St_{trans}, find a record $\langle i, \mathsf{transcript}_i = (X_i, \mathsf{ID}(v_i), C_{v_i}, \sigma_{v_i}, sig_i)\rangle$ such that $X_i = \tilde{X}$. If no such record exists in St_{trans}, return $\perp$. Otherwise, return i.

Each signature now consists of 150 group elements since **com** and $\mathbf{\Pi}$ contain 69 and 63 group elements, respectively. The only overhead is in the size of the group public key which grows from $O(\log N)$ to $O(\log^2 N)$.

C.3 Security

Theorem 5 (Misidentification). *The scheme is secure against misidentification attacks assuming that the q-SFP and the FSDH problems are both hard for* $q = \max(q_a, q_r^2)$*, where* q_a *and* q_r *denote the maximal numbers of* $Q_{\mathsf{a\text{-}join}}$ *queries and* Q_{revoke} *queries, respectively, and* N *is the maximal number of group members.*

Proof. The proof is almost identical to the proof of Theorem 2. It considers the same two kinds of forgeries and the only difference is the treatment of Type II.b forgeries. Lemma 3 shows how to break the 2-3-SqDH assumption using a Type II.b forger. □

Lemma 3. *The advantage of any Type II.b forger* $\mathcal{A}$ *is at most* $\mathbf{Adv}_{\mathcal{A}}^{\text{mis-id-II.b}}(\lambda) \leq \ell \cdot \mathbf{Adv}^{\text{FSDH}}(\lambda)$*, where* $\ell = \log N$ *and* N *is the maximal number of users.*

Proof. To prove the result, it is convenient to use an equivalent formulation[3] of the problem. Namely, given (g, g^a), we have to find a triple $(g^{a\cdot\mu}, g^{\mu}, g^{\mu/a})$ for some $\mu \neq 0$. We describe an algorithm $\mathcal{B}$ that receives as input an instance $(g, g^a) \in \mathbb{G}^2$ of the FSDH problem and uses the Type II.b forger to find a non-trivial $(g^{a\cdot\mu}, g^{\mu}, g^{\mu/a})$. To generate the group public key, $\mathcal{B}$ follows the specification of the Setup procedure except that, instead of computing ck as in step 3 of the algorithm, it defines $ck = (g_1, \ldots, g_\ell, h_1, \ldots, h_\ell, \{h_{i,j}\}_{i \neq j})$ as follows. It picks $i^\star \xleftarrow{R} \{1, \ldots, \ell\}$ and defines

$$\begin{array}{rlr}
g_{i^\star} &= g^{a} & \\
g_i &= g^{z_i} & i \neq i^\star \\
h_{i^\star} &= g^{\prod_{\kappa \neq i^\star} z_\kappa} & \\
h_i &= (g^a)^{\prod_{\kappa \neq i, i^\star} z_\kappa} & i \neq i^\star \\
h_{ij} &= (g^a)^{\prod_{\kappa \neq i,j,i^\star} z_\kappa} & i \neq i^\star,\ j \neq i^\star \\
h_{i^\star j} &= g^{\prod_{\kappa \neq j, i^\star} z_\kappa} & j \neq i^\star \\
h_{i i^\star} &= g^{\prod_{\kappa \neq i, i^\star} z_\kappa} & i \neq i^\star
\end{array}$$

where $z_1, \ldots, z_\ell \xleftarrow{R} \mathbb{Z}_p$. Eventually, $\mathcal{Y} := \big(g, pk_{\mathsf{AHO}}^{(0)}, pk_{\mathsf{AHO}}^{(1)}, ck, \mathbf{f}, \vec{\varphi}, (U, V), \Sigma\big)$ is given to the Type II.b forger $\mathcal{A}$.

During the whole game, the adversary can adaptively probe the Q_{pub}, $Q_{\mathsf{a\text{-}join}}$, Q_{revoke}, Q_{read}, and Q_{keyOA} oracles. Since $\mathcal{S}_{\mathsf{GM}} = (sk_{\mathsf{AHO}}^{(0)}, sk_{\mathsf{AHO}}^{(1)})$ and $\mathcal{S}_{\mathsf{OA}} = (\beta_1, \beta_2)$ are available to the reduction $\mathcal{B}$, the latter can always perfectly answer adversarial queries. At the end of the game, the adversary $\mathcal{A}$ outputs a forgery $\sigma^\star$ for which the committed variables $C_{v_i}^\star$, $(R_{l,2}^\star, \ldots, R_{l,7}^\star)$, $(\Psi_{l,0}^\star, \Psi_{l,1}^\star)$ and $(\Gamma_l^\star, W_{\phi_l}^\star, W_{\psi_l}^\star)$ satisfy the relations

$$e(g, C_{v_i}^\star) \cdot e(R_{l,4}^\star, R_{l,3}^\star) = e(R_{l,2}^\star, W_{\phi_l}^\star) \tag{32}$$

$$e(g, C_{v_i}^\star) \cdot e(\Psi_{l,0}^\star, R_{l,6}^\star) = e(R_{l,5}^\star, W_{\psi_l}^\star), \tag{33}$$

$$e(R_{l,7}^\star \cdot \Psi_{l,0}^\star, \Gamma_l^\star) = e(g, g), \tag{34}$$

$$e(\Psi_{l,0}^\star, R_{l,5}^\star) = e(g, \Psi_{l,1}^\star). \tag{35}$$

[3] Given (g, g^a), if we define $y = g^a$ and $y^A = g$ (so that $A = 1/a$) any FSDH solution $(y^\mu, y^{A\cdot\mu}, y^{(A^2)^\mu})$ can be written as $(g^{a\cdot\mu}, g^\mu, g^{\mu/a})$

although $\sigma^\star$ opens to some user $i^\star \in U^a \cap \mathcal{R}_{t^\star}$.

Note that $(R^\star_{l,1}, \ldots, R^\star_{l,7})$ is necessarily of the form

$$(R^\star_{l,1}, R^\star_{l,2}, R^\star_{l,3}, R^\star_{l,4}, R^\star_{l,5}, R^\star_{l,6}, R^\star_{l,7}) = \Big(g^{t^\star},\ g_{\phi_l},\ h_{\phi_l},\ g^{-\mathsf{ID}(x^\star_{k_l})},\ g_{\psi_l},\ h_{\psi_l},\ g^{\mathsf{ID}(x^\star_{k_l})}\Big), \tag{36}$$

for some indices $\phi_l, \psi_l \in \{1, \ldots, \ell\}$ and some node identifiers $\mathsf{ID}(x^\star_{k_l}), \mathsf{ID}(x^\star_{u_l}) \in \{1, \ldots, 2N-1\}$ that were chosen by $\mathcal{B}$ at the latest Q_{revoke}-query. Since, by hypothesis, $\sigma^\star$ contains a committed pair $(X^\star, C^\star_{v_i})$ that was signed by $\mathcal{B}$ during some $Q_{\mathsf{a\text{-}join}}$-query, $\mathcal{B}$ also knows $(I^\star_1, \ldots, I^\star_\ell)$ such that $C^\star_{v_i} = \prod_{\kappa=1}^{\ell} g_\kappa^{I^\star_\kappa}$. Since $i^\star \in U^a \cap \mathcal{R}_{t^\star}$, it must hold that either:

- $I^\star_{\phi_l} \neq \mathsf{ID}(x^\star_{k_l})$: In this case, relations (36) and (32) imply that

$$e(g, C^\star_{v_i}) \cdot e(g^{-\mathsf{ID}(x^\star_{k_l})}, h_{\phi_\ell}) = e(g_{\phi_l}, W^\star_{\phi_l}) \tag{37}$$

for values $\phi_l \in \{1, \ldots, \ell\}$ and $\mathsf{ID}(x_{k_l})^\star \in \{1, \ldots, 2N-1\}$ that are available to $\mathcal{B}$. At this point, $\mathcal{B}$ fails if $\phi_l \neq i^\star$. With probability $1/\ell$ however, it holds that $\phi_l = i^\star$ in which case $\mathcal{B}$ can solve the problem as follows. Since it knows $(I^\star_1, \ldots, I^\star_\ell)$ such that $C^\star_{v_i} = \prod_{\kappa=1}^{\ell} g_\kappa^{I^\star_\kappa}$, it can compute $W' = \prod_{\kappa=1,\ \kappa\neq\phi_l}^{\ell} h_{\phi_l,\kappa}^{I^\star_\kappa}$ which satisfies

$$e(g, C^\star_{v_i}) \cdot e(g^{-I^\star_{\phi_l}}, h_{\phi_\ell}) = e(g_{\phi_l}, W'_{\phi_l}) \tag{38}$$

By dividing (37) and (38), we find that $e(g_{i^\star}, (W^\star_{\phi_l}/W'_{\phi_l})^{1/(I^\star_{\phi_l} - \mathsf{ID}(x^\star_{k_l}))}) = e(g, h_{i^\star})$. This implies that, by computing $g^{1/a} = \big(W^\star_{\phi_l}/W'\big)^{1/(I^\star_{\phi_l} - \mathsf{ID}(x^\star_{k_l}))\cdot\prod_{\kappa\neq i^\star} z_\kappa}$, $\mathcal{B}$ actually solves a problem which is at least as hard as FSDH.

- $I^\star_{\psi_l} = \mathsf{ID}(x^\star_{u_l})$: If we define $\varrho = -\log_g(\Psi^\star_{l,0})$, relations (36) and (33)-(35) imply that

$$e(g, C^\star_{v_i}) \cdot e(g^{-\varrho}, h_{\psi_\ell}) = e(g_{\psi_l}, W'_{\psi_l}) \tag{39}$$

$$g^{\varrho - I^\star_{\psi_l}} \neq 1_{\mathbb{G}} \tag{40}$$

$$\Psi^\star_{l,0} = g^{-\varrho} \tag{41}$$

$$\Psi^\star_{l,1} = g_{\psi_l}^{-\varrho}, \tag{42}$$

for some $\psi_l \in \{1, \ldots, \ell\}$. At this point, $\mathcal{B}$ halts and declares failure if $\psi_l \neq i^\star$. Still, with probability $1/\ell$, we have $\psi_l = i^\star$ and $\mathcal{B}$ can solve the 2-3-SqDH as follows. Similarly to the previous case, it can compute $W' = \prod_{\kappa=1,\ \kappa\neq\psi_l}^{\ell} h_{\psi_l,\kappa}^{I^\star_\kappa}$ such that

$$e(g, C^\star_{v_i}) \cdot e(g^{-I^\star_{\psi_l}}, h_{\psi_\ell}) = e(g_{\psi_l}, W^\star_{\psi_l}) \tag{43}$$

Now, by dividing (39) from (43), we obtain the equality $e(g, h_{\psi_l})^{\varrho - I^\star_{\psi_l}} = e(g_{\psi_l}, W'/W^\star_{\psi_l})$ which, if $\psi = i^\star$, implies $W'/W^\star_{\phi_l} = g^{(\varrho - I^\star_{\psi_l})\cdot\prod_{\kappa\neq i^\star} z_\kappa/a}$. The triple

$$\begin{aligned}&\Big(({\Psi^\star_{l,1}}^{-1} \cdot (g^a)^{-I^\star_{\psi_l}})^{\prod_{\kappa\neq i^\star} z_\kappa},\ ({\Psi^\star_{l,0}}^{-1} \cdot g^{-I^\star_{\psi_l}})^{\prod_{\kappa\neq i^\star} z_\kappa},\ W'/W^\star_{\phi_l}\Big)\\ &\qquad = \Big(g^{a\left(\varrho - I^\star_{\psi_l}\right)\cdot\prod_{\kappa\neq i^\star} z_\kappa},\ g^{\left(\varrho - I^\star_{\psi_l}\right)\cdot\prod_{\kappa\neq i^\star} z_\kappa},\ g^{\frac{(\varrho - I^\star_{\psi_l})\cdot\prod_{\kappa\neq i^\star} z_\kappa}{a}}\Big)\end{aligned}$$

is a non-trivial solution to the FSDH instance.

In both cases, we observe that, if $\mathcal{A}$ is able to mount a Type II.b attack with probability ε, then $\mathcal{B}$ is able to break the Flexible Square Diffie-Hellman assumption with probability ε/ℓ. $\square$

The proofs of anonymity and security against framing attacks are identical to those of the first scheme and omitted here.

C.4 Further Reducing the Number of Assumptions

We note that, using the technique of Malkin, Teranishi, Vahlis and Yung [48], it is possible to replace the SDH assumption by the standard Diffie-Hellman assumption in the proof of security against framing attack. To this end, we must introduce a Waters-like [60] number theoretic hash function (described by $O(\lambda)$ group elements) in the group public key in order to have a message-dependent Groth-Sahai CRS. Namely, all proofs of the signature are generated w.r.t. a Groth-Sahai CRS $(\vec{f}_1, \vec{f}_2, \vec{f}_{\mathsf{VK}})$, where $\vec{f}_{\mathsf{VK}}$ is obtained by "hashing" the verification key of a one-time signature. In order to secure the scheme against framing attacks, each group signature should prove knowledge of a value (such as $g^{1/x}$, where $x = \log_g(X)$) that only the signer knows. Finally, all non-interactive proofs should be signed along with the actual message using the private key SK of the one-time signature[4].

The details are omitted here but it is not hard to see that a successful framing attack would imply a PPT algorithm to compute $g^{1/x}$ given $X = g^x$, which is equivalent to solving the Diffie-Hellman problem. Eventually, we only need the q-SFP assumption, the FSDH assumption and the DLIN assumption to prove the security of the scheme. In the resulting group signature, the group public key is larger and comprises $O(\lambda + \log^2 N)$ group elements.

[4] The reason why $\vec{f}_{\mathsf{VK}}$ is not directly derived from M is that we need to prevent Groth-Sahai proofs from being publicly randomized in order to achieve anonymity in the CCA2 sense: as noted in [35], signatures should not be re-randomizable in order to attain anonymity in the strongest sense.

Linearly Homomorphic Structure-Preserving Signatures and Their Applications

Benoît Libert[1], Thomas Peters[2*], Marc Joye[1], and Moti Yung[3]

[1] Technicolor (France)
[2] Université catholique de Louvain, Crypto Group (Belgium)
[3] Google Inc. and Columbia University (USA)

Abstract. Structure-preserving signatures (SPS) are signature schemes where messages, signatures and public keys all consist of elements of a group over which a bilinear map is efficiently computable. This property makes them useful in cryptographic protocols as they nicely compose with other algebraic tools (like the celebrated Groth-Sahai proof systems). In this paper, we consider SPS systems with homomorphic properties and suggest applications that have not been provided before (in particular, not by employing ordinary SPS). We build linearly homomorphic structure-preserving signatures under simple assumptions and show that the primitive makes it possible to verify the calculations performed by a server on outsourced encrypted data (*i.e.*, combining secure computation and authenticated computation to allow reliable and secure cloud storage and computation, while freeing the client from retaining cleartext storage). Then, we give a generic construction of non-malleable (and actually simulation-sound) commitment from any linearly homomorphic SPS. This notably provides the first constant-size non-malleable commitment to group elements.

Keywords: Structure-preserving cryptography, signatures, homomorphism, commitment schemes, non-malleability.

1 Introduction

Composability is an important cryptographic design notion for building systems and protocols. Inside protocols, cryptographic tools need to compose well with each other in order to be used in combination. Structure-preserving cryptography [3], in turn, is a recent paradigm that takes care of composing algebraic tools, and primarily within groups supporting bilinear maps to allow smooth composition with the Groth-Sahai proof systems [47]. The notion allows for modular and simplified designs of various cryptographic protocols and primitives. In the last three years, a large body of work has analyzed the feasibility and the efficiency of structure-preserving signatures (SPS) [45, 28, 38, 1, 3, 4, 20, 29, 51, 5, 6], public-key encryption [21] and commitments schemes [48, 2].

In this paper, we consider SPS schemes with linearly homomorphic properties and argue that such primitives have many applications, even independently of Groth-Sahai proofs. Let us next review our results and then review related work.

1.1 Our Contributions

Linearly Homomorphic Structure-Preserving Signatures. In this paper, we put forth the notion of linearly homomorphic structure-preserving signatures (linearly homomorphic signatures and structure-preserving signatures have been defined before, as we review in the sequel, but the combination of the earlier notions is useful and non-trivial). These signature schemes function exactly like ordinary homomorphic signatures with the additional restriction that signatures and messages only consist of (vectors of) group elements whose discrete logarithms may not be available. We describe three constructions and prove their security under established complexity assumptions in symmetric bilinear groups.

[*] This author was supported by the CAMUS Walloon Region Project.

Applications. As in all SPS systems, the structure-preserving property makes it possible to efficiently prove knowledge of a homomorphic signature on a committed vector. However, as indicated above, we describe applications of linearly homomorphic SPS beyond their compatibility with the Groth-Sahai techniques.

First, we show that the primitive enables verifiable computation mechanisms on encrypted data.[4] Specifically, it allows a client to store encrypted files on an untrusted remote server. While the dataset is encrypted using an additively homomorphic encryption scheme, the server is able to blindly compute linear functions on the original data and provide the client with a short homomorphically derived signature vouching for the correctness of the computation. This is achieved by having the client sign each ciphertext using a homomorphic SPS scheme and handing the resulting signatures to the server at the beginning. After this initial phase, the client only needs to store a short piece of information, no matter how large the file is. Still, he remains able to authenticate linear functions on his data and the whole process is completely non-interactive. The method extends when datasets are encrypted using a CCA1-secure encryption schemes. Indeed, we will observe that linearly homomorphic SPS schemes yield simple homomorphic IND-CCA1-secure cryptosystems with publicly verifiable ciphertexts.

As a second and perhaps more surprising application, we show that linearly homomorphic SPS schemes generically yield non-malleable [35] trapdoor commitments to group elements. We actually construct a simulation-sound trapdoor commitment [40] — a primitive known (by [40, 54]) to imply re-usable non-malleable commitments with respect to opening [31] — from any linearly homomorphic SPS satisfying a relatively mild condition. To our knowledge, we thus obtain the first constant-size trapdoor commitments to group elements providing re-usable non-malleability with respect to opening. Previous non-interactive commitments to group elements were either malleable [47, 48] or inherently length-increasing [36]: if we disregard the trivial solution consisting of hashing the message first (which is not an option when we want to allow for efficient proofs of knowledge of an opening), no general technique has been known, to date, for committing to many group elements at once using a short commitment string.

In the structure-preserving case, our transformation is purely generic as it applies to a template which any linearly homomorphic SPS necessarily satisfies in symmetric bilinear groups. We also generalize the construction so as to build simulation-sound trapdoor commitments to vectors from any pairing-based (non-structure-preserving) linearly homomorphic signature. In this case, the conversion is only semi-generic as it imposes conditions which are only met by pairing-based systems for the time being: essentially, we need the underlying signature scheme to operate over groups of finite, public order. While only partially generic, this construction of non-malleable commitments from linearly homomorphic signatures is somewhat unexpected considering that the terms “non-malleability” and “homomorphism” are antagonistic, and thus may be considered incompatible.

Techniques and ideas. At first, the very name of our primitive may sound almost self-contradictory when it comes to formally define its security. Indeed, the security of a linearly homomorphic scheme [17] notably requires that it be infeasible to publicly compute a signature on a vector outside the linear span of originally signed vectors. The problem is that, when vector entries live in a discrete-logarithm hard group, deciding whether several vectors are independent or not is believed to be a hard problem. Yet, this will not prevent us from applying new techniques and constructing schemes with security proofs under simple assumptions and the reduction will be able to detect when the adversary has won by simply solving the problem instance it received as input.

Our first scheme's starting point is the one-time (regular) SPS scheme of Abe *et al.* [1]. By removing certain public key components, we obtain the desired linear homomorphism, and prove the security using information-theoretic arguments as in [1]. The key observation here is that, as long as the adversary does not output a signature on a linear combination of previously signed vectors,

[4] Our goals are very different from those of [42], where verifiable computation on homomorphically encrypted data is also considered. We do not seek to outsource computation but rather save the client from storing large datasets.

it will be unable to sign its target vector in the same way as the reduction would, because certain private key components will remain perfectly hidden.

Our initial scheme inherits the one-time restriction of the scheme in [1] in that only one linear subspace can be safely signed with a given public key. Nevertheless, we can extend it to build a full linearly homomorphic SPS system. To this end, we suitably combine our first scheme with Waters signatures [60]. Here, Waters signatures are used as a resting ground for fresh random exponents which are introduced in each signed vector and help us refresh the state of the system and apply each time the same argument as in the one-time scheme. We also present techniques to turn the scheme into a fully randomizable one, where a derived signature has the same distribution as a directly signed message.

In our simulation-sound commitments to group elements, the commitment generation technique appeals to the verification algorithm of the signature scheme, and proceeds by evaluating the corresponding pairing-product equations on the message, but using random group elements instead of actual signatures. The binding and simulation-binding properties, in turn, stem from the infeasibility of forging signatures while the signature homomorphism allows equivocating fake commitments when simulating the view of an adversary. It was already known how to build simulation-sound and non-malleable commitments [40, 54, 31, 41, 24] from signature schemes with efficient Σ protocols. Our method is, in fact, different and immediately yields length-reducing structure-preserving commitments to vectors without using Σ protocols.

1.2 Related Work

Structure-Preserving Signatures. Signature schemes where messages only consist of group elements appeared for the first time — without the "structure-preserving" terminology — as ingredients of Groth's construction [45] of group signatures in the standard model. The scheme of [45] was mostly a proof of concept, with signatures consisting of thousands of group elements. More efficient realizations were given by Cathalo, Libert and Yung [28] and Fuchsbauer [38]. Abe, Haralambiev and Ohkubo [1, 3] subsequently showed how to sign messages of n group elements at once using $O(1)$-size signatures. Lower bounds on the size of structure-preserving signatures were given in [4] while Abe *et al.* [7] provided evidence that optimally short SPS necessarily rely on interactive assumptions. As an ingredient for their tightly secure cryptosystems, Hofheinz and Jager [51] gave constructions based on the Decision Linear assumption [16] while similar results were independently achieved in [20, 29]. Quite recently, Abe *et al.* [5, 6] obtained constant-size signatures without sacrificing the security guarantees offered by security proofs under simple assumptions.

Regarding primitives beyond signature schemes, Camenisch *et al.* [21] showed a structure-preserving variant of the Cramer-Shoup cryptosystem [30] and used it to implement oblivious third parties [22]. Groth [48] described length-reducing trapdoor commitments (*i.e.*, where the commitment is shorter than the committed message) to group elements whereas [2] showed the impossibility of realizing such commitments when the commitment string lives in the same group as the message. Sakai *et al.* [58] recently suggested to use structure-preserving identity-based encryption [59] systems to restrict the power of the opening authority in group signatures.

Linearly Homomorphic Signatures. The concept of homomorphic signatures can be traced back to Desmedt [33] while proper definitions remained lacking until the work of Johnson *et al.* [53]. Since then, constructions have appeared for various kinds of homomorphisms (see [8] and references therein).

Linearly homomorphic signatures are an important class of homomorphic signatures for arithmetic functions, whose study was initiated by Boneh, Freeman, Katz and Waters [17]. While initially motivated by applications to network coding [17], they are also useful in proofs of storage [9, 10] or in verifiable computation mechanisms, when it comes to authenticate servers' computations on outsourced data (see, *e.g.*, [8]). The recent years, much attention was given to the notion and a variety of constructions [43, 11, 18, 19, 26, 27, 37, 12, 13] based on various assumptions have been studied.

1.3 Organization

Section 2 first gives security definitions for linearly homomorphic SPS systems, for which efficient constructions are provided in Section 3. Their applications to verifiable computation on encrypted data are explained in Section 4 while Section 5 shows how to build simulation-sound commitments to group elements. Implications and generalizations of the latter are then given in Appendix E.

2 Background

2.1 Definitions for Linearly Homomorphic Signatures

Let $(\mathbb{G}, \mathbb{G}_T)$ be a configuration of (multiplicatively written) groups of prime order p over which a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is efficiently computable.

Following [1, 3], we say that a signature scheme is *structure-preserving* if messages, signature components and public keys live in the group $\mathbb{G}$.

We consider linearly homomorphic signatures for which the message space $\mathcal{M}$ consists of pairs $\mathcal{M} := \mathcal{T} \times \mathbb{G}^n$, for some $n \in \mathbb{N}$, where $\mathcal{T}$ is a tag space. We remark that, in the applications considered in this paper, tags do not need to be group elements. We thus allow them to be arbitrary strings.

Definition 1. *A linearly homomorphic structure-preserving signature scheme over $(\mathbb{G}, \mathbb{G}_T)$ consists of a tuple of efficient algorithms $\Sigma = (\mathsf{Keygen}, \mathsf{Sign}, \mathsf{SignDerive}, \mathsf{Verify})$ for which the message space is $\mathcal{M} := \mathcal{T} \times \mathbb{G}^n$, for some $n \in \mathsf{poly}(\lambda)$ and some set $\mathcal{T}$, and with the following specifications.*

$\mathsf{Keygen}(\lambda, n)$: *is a randomized algorithm that takes in a security parameter $\lambda \in \mathbb{N}$ and an integer $n \in \mathsf{poly}(\lambda)$ denoting the dimension of vectors to be signed. It outputs a key pair $(\mathsf{pk}, \mathsf{sk})$ and the description of a tag (i.e., a file identifier) space $\mathcal{T}$.*

$\mathsf{Sign}(\mathsf{sk}, \tau, \vec{M})$: *is a possibly probabilistic algorithm that takes as input a private key sk, a file identifier $\tau \in \mathcal{T}$ and a vector $\vec{M} \in \mathbb{G}^n$. It outputs a signature $\sigma \in \mathbb{G}^{n_s}$, for some $n_s \in \mathsf{poly}(\lambda)$.*

$\mathsf{SignDerive}(\mathsf{pk}, \tau, \{(\omega_i, \sigma^{(i)})\}_{i=1}^{\ell})$: *is a (possibly probabilistic) signature derivation algorithm. It takes as input a public key pk, a file identifier τ as well as ℓ pairs $(\omega_i, \sigma^{(i)})$, each of which consists of a weight $\omega_i \in \mathbb{Z}_p$ and a signature $\sigma^{(i)} \in \mathbb{G}^{n_s}$. The output is a signature $\sigma \in \mathbb{G}^{n_s}$ on the vector $\vec{M} = \prod_{i=1}^{\ell} \vec{M}_i^{\,\omega_i}$, where $\sigma^{(i)}$ is a signature on $\vec{M}_i$.*

$\mathsf{Verify}(\mathsf{pk}, \tau, \vec{M}, \sigma)$: *is a deterministic algorithm that takes in a public key pk, a file identifier $\tau \in \mathcal{T}$, a signature σ and a vector $\vec{M}$. It outputs 1 if σ is deemed valid and 0 otherwise.*

Correctness is expressed by imposing that, for all security parameters $\lambda \in \mathbb{N}$, all integers $n \in \mathsf{poly}(\lambda)$ and all triples $(\mathsf{pk}, \mathsf{sk}, \mathcal{T}) \leftarrow \mathsf{Keygen}(\lambda, n)$, the following holds:

1. For all $\tau \in \mathcal{T}$ and all n-vectors $\vec{M}$, if $\sigma = \mathsf{Sign}(\mathsf{sk}, \tau, \vec{M})$, then we have $\mathsf{Verify}(\mathsf{pk}, \tau, \vec{M}, \sigma) = 1$.
2. For all $\tau \in \mathcal{T}$, any $\ell > 0$ and any set of triples $\{(\omega_i, \sigma^{(i)}, \vec{M}_i)\}_{i=1}^{\ell}$, if $\mathsf{Verify}(\mathsf{pk}, \tau, \vec{M}_i, \sigma^{(i)}) = 1$ for each $i \in \{1, \ldots, \ell\}$, then $\mathsf{Verify}\big(\mathsf{pk}, \tau, \prod_{i=1}^{\ell} \vec{M}_i^{\,\omega_i}, \mathsf{SignDerive}(\mathsf{pk}, \tau, \{(\omega_i, \sigma^{(i)})\}_{i=1}^{\ell})\big) = 1$.

Security. In linearly homomorphic signatures, we use the same definition of unforgeability as in [12]. This definition implies security in the stronger model used by Freeman [37] since the adversary can interleave signing queries for individual vectors belonging to distinct subspaces. Moreover, file identifiers can be chosen by the adversary (which strengthens the definition of [17]) and are not assumed to be uniformly distributed. As a result, a file identifier can be a low-entropy, easy-to-remember string such as the name of the dataset's owner.

Definition 2. *A linearly homomorphic SPS scheme $\Sigma = (\mathsf{Keygen}, \mathsf{Sign}, \mathsf{Verify})$ is secure if no PPT adversary has non-negligible advantage in the game below:*

1. *The adversary $\mathcal{A}$ chooses an integer $n \in \mathbb{N}$ and sends it to the challenger who runs $\mathsf{Keygen}(\lambda, n)$ and obtains $(\mathsf{pk}, \mathsf{sk})$ before sending pk to $\mathcal{A}$.*
2. *On polynomially-many occasions, $\mathcal{A}$ can interleave the following kinds of queries.*
   - *Signing queries: $\mathcal{A}$ chooses a tag $\tau \in \mathcal{T}$ and a vector $\vec{M} \in \mathbb{G}^n$. The challenger picks a handle h and computes $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \tau, \vec{M})$. It stores $(\mathsf{h}, (\tau, \vec{M}, \sigma))$ in a table T and returns h.*
   - *Derivation queries: $\mathcal{A}$ chooses a vector of handles $\vec{\mathsf{h}} = (\mathsf{h}_1, \ldots, \mathsf{h}_k)$ and a set of coefficients $\{\omega_i\}_{i=1}^k$. The challenger retrieves the tuples $\{(\mathsf{h}_i, (\tau, \vec{M}_i), \sigma^{(i)})\}_{i=1}^k$ from T and returns $\perp$ if one of these does not exist or if there exists $i \in \{1, \ldots, k\}$ such that $\tau_i \neq \tau$. Otherwise, it computes $\vec{M} = \prod_{i=1}^k \vec{M}_i^{\omega_i}$ and runs $\sigma' \leftarrow \mathsf{SignDerive}(\mathsf{pk}, \tau, \{(\omega_i, \sigma^{(i)})\}_{i=1}^k)$. It also chooses a handle h', stores $(\mathsf{h}', (\tau, \vec{M}), \sigma')$ in T and returns h' to $\mathcal{A}$.*
   - *Reveal queries: $\mathcal{A}$ chooses a handle h. If no tuple of the form $(\mathsf{h}, (\tau, \vec{M}), \sigma')$ exists in T, the challenger returns $\perp$. Otherwise, it returns σ' to $\mathcal{A}$ and adds $((\tau, \vec{M}), \sigma')$ to the set Q.*
3. *$\mathcal{A}$ outputs an identifier $\tau^\star$, a signature $\sigma^\star$ and a vector $\vec{M}^\star \in \mathbb{G}^n$. The adversary $\mathcal{A}$ wins if $\mathsf{Verify}(\mathsf{pk}, \tau^\star, \vec{M}^\star, \sigma^\star) = 1$ and one of the conditions below is satisfied:*
   - ○ *(Type I): $\tau^\star \neq \tau_i$ for any entry $(\tau_i, .)$ in Q and $\vec{M}^\star \neq (1_{\mathbb{G}}, \ldots, 1_{\mathbb{G}})$.*
   - ○ *(Type II): $\tau^\star = \tau_i$ for $k_i > 0$ entries $(\tau_i, .)$ in Q and $\vec{M}^\star \notin V_i$, where V_i denotes the subspace spanned by all vectors $\vec{M}_1, \ldots, \vec{M}_{k_i}$ for which an entry of the form $(\tau^\star, \vec{M}_j)$, with $j \in \{1, \ldots, k_i\}$, appears in Q.*

$\mathcal{A}$'s advantage is its probability of success taken over all coin tosses.

In our first scheme, we will consider a weaker notion of *one-time* security. In this notion, the adversary is limited to obtain signatures for only *one* linear subspace. In this case, there is no need for file identifiers and we assume that all vectors are assigned the identifier $\tau = \varepsilon$.

In the following, the adversary will be said *independent* if

- For any given tag τ, it is restricted to only query signatures on linearly independent vectors.
- Each vector is only queried at most once.

Non-independent adversaries are not subject to the above restrictions. It will be necessary to consider these adversaries in our construction of non-malleable commitments. Nevertheless, security against independent adversaries suffices for many applications — including encrypted cloud storage — since the signer can always append unit vectors to each newly signed vector.

At first, one may wonder how Definition 2 can be satisfied at all given that the challenger may not have an efficient way to check whether the adversary is successful. Indeed, in cryptographically useful discrete-logarithm-hard groups $\mathbb{G}$, deciding whether vectors $\{\vec{M}_i\}_i$ of $\mathbb{G}^n$ are linearly dependent is believed to be difficult when $n > 2$. However, it may be possible using some trapdoor information embedded in pk, especially if the adversary additionally outputs signatures on $\{\vec{M}_i\}_i$.

In some applications, it makes sense to consider a weaker attack model where, in the case of Type II attacks, the adversary is only deemed successful if it can output a convincing proof that its target vector $\vec{M}^\star$ is indeed independent of the vectors that were signed for the tag $\tau^\star$. The proof can be either a NIZK proof or, alternatively, a vector in the kernel of the matrix whose rows are the vectors that were signed for $\tau^\star$. We call such an adversary a *targeting* adversary.

2.2 Hardness Assumptions

We rely on the following hardness assumptions, the first of which implies the second one.

Definition 3 ([16]). *The* Decision Linear Problem *(DLIN) in $\mathbb{G}$, is to distinguish the distributions $(g^a, g^b, g^{ac}, g^{bd}, g^{c+d})$ and $(g^a, g^b, g^{ac}, g^{bd}, g^z)$, with $a, b, c, d \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$, $z \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$. The* Decision Linear Assumption *is the intractability of DLIN for any PPT distinguisher $\mathcal{D}$.*

Definition 4. *The* Simultaneous Double Pairing problem *(SDP) in* $(\mathbb{G}, \mathbb{G}_T)$ *is, given a tuple of elements* $(g_z, g_r, h_z, h_u) \in_R \mathbb{G}^4$*, to find a non-trivial triple* $(z, r, u) \in \mathbb{G}^3 \backslash \{(1_{\mathbb{G}}, 1_{\mathbb{G}}, 1_{\mathbb{G}})\}$ *such that* $e(g_z, z) \cdot e(g_r, r) = 1_{\mathbb{G}_T}$ *and* $e(h_z, z) \cdot e(h_u, u) = 1_{\mathbb{G}_T}$*.*

3 Constructions of Linearly Homomorphic Structure-Preserving Signatures

As a warm-up, we begin by describing a one-time homomorphic signature, where a given public key allows signing only *one* linear subspace.

3.1 A One-Time Linearly Homomorphic Construction

In the description hereunder, since only one linear subspace can be signed for each public key, no file identifier τ is used. We thus set τ to be the empty string ε in all algorithms.

Keygen(λ, n): given a security parameter λ and the dimension $n \in \mathbb{N}$ of the subspace to be signed, choose bilinear group $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$. Then, choose generators $h, g_z, g_r, h_z \stackrel{R}{\leftarrow} \mathbb{G}$. Pick $\chi_i, \gamma_i, \delta_i \stackrel{R}{\leftarrow} \mathbb{Z}_p$, for $i = 1$ to n. Then, for each $i \in \{1, \ldots, n\}$, compute $g_i = g_z^{\chi_i} g_r^{\gamma_i}$, $h_i = h_z^{\chi_i} h^{\delta_i}$. The private key is $\mathsf{sk} = \{\chi_i, \gamma_i, \delta_i\}_{i=1}^n$ while the public key is defined to be

$$\mathsf{pk} = \big(g_z,\ h_r,\ h_z,\ h,\ \{g_i, h_i\}_{i=1}^n\big) \in \mathbb{G}^{2n+4}.$$

Sign(sk, τ, ($M_1, \ldots, M_n$)): to sign a vector $(M_1, \ldots, M_n) \in \mathbb{G}^n$ associated with the identifier $\tau = \varepsilon$ using $sk = \{\chi_i, \gamma_i, \delta_i\}_{i=1}^n$, compute the signature consists of $\sigma = (z, r, u) \in \mathbb{G}^3$, where

$$z = \prod_{i=1}^n M_i^{-\chi_i}, \qquad r = \prod_{i=1}^n M_i^{-\gamma_i}, \qquad u = \prod_{i=1}^n M_i^{-\delta_i}.$$

SignDerive(pk, τ, $\{(\omega_i, \sigma^{(i)})\}_{i=1}^\ell$): given the public key pk, a file identifier $\tau = \varepsilon$ and ℓ tuples $(\omega_i, \sigma^{(i)})$, parse each signature $\sigma^{(i)}$ as $\sigma^{(i)} = \big(z_i, r_i, u_i\big) \in \mathbb{G}^3$ for $i = 1$ to ℓ. Compute and return the derived signature $\sigma = (z, r, u) = \big(\prod_{i=1}^\ell z_i^{\omega_i}, \prod_{i=1}^\ell r_i^{\omega_i}, \prod_{i=1}^\ell u_i^{\omega_i}\big)$.

Verify(pk, σ, τ, ($M_1, \ldots, M_n$)): given a signature $\sigma = (z, r, u) \in \mathbb{G}^3$, a vector $(M_1, \ldots, M_n)$ and a file identifier $\tau = \varepsilon$, return 1 if and only if $(M_1, \ldots, M_n) \neq (1_{\mathbb{G}}, \ldots, 1_{\mathbb{G}})$ and (z, r, u) satisfy

$$1_{\mathbb{G}_T} = e(g_z, z) \cdot e(g_r, r) \cdot \prod_{i=1}^n e(g_i, M_i), \qquad 1_{\mathbb{G}_T} = e(h_z, z) \cdot e(h, u) \cdot \prod_{i=1}^n e(h_i, M_i).$$

The proof of security relies on the fact that, while the signing algorithm is deterministic, signatures are not unique. However, the reduction will be able to compute exactly one signature for each vector. At the same time, an adversary has no information about which specific signature the legitimate signer would compute on a vector outside the span of already signed vectors. Moreover, by obtaining two distinct signatures on a given vector, the reduction can solve a given SDP instance.

Theorem 1. *The scheme is unforgeable if the SDP assumption holds in* $(\mathbb{G}, \mathbb{G}_T)$*.*

Proof. We describe an algorithm $\mathcal{B}$ that takes as input a SDP instance $(g_z, g_r, h_z, h) \in \mathbb{G}^4$ and uses a forger $\mathcal{A}$ to find a triple (z, r, u) such that $e(g_z, z) \cdot e(g_r, r) = e(h_z, z) \cdot e(h, u) = 1_{\mathbb{G}_T}$.

To this end, $\mathcal{B}$ honestly runs the key generation algorithm using randomly chosen $\{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^n$. Whenever $\mathcal{A}$ requests a signature on a vector $(M_1, \ldots, M_n) \in \mathbb{G}^n$, $\mathcal{B}$ faithfully follows the specification of the signing algorithm. The game ends with the adversary $\mathcal{A}$ outputting a vector $(M_1^\star, \ldots, M_n^\star)$ with a valid signature $(z^\star, r^\star, u^\star)$. At this point, $\mathcal{B}$ computes its own signature

$$(z^\dagger, r^\dagger, u^\dagger) = (\prod_{i=1}^n M_i^{\star-\chi_i}, \prod_{i=1}^n M_i^{\star-\gamma_i}, \prod_{i=1}^n M_i^{\star-\delta_i}) \tag{1}$$

on $(M_1^\star, \dots, M_n^\star)$. We claim that, with overwhelming probability, $(z^\ddagger, r^\ddagger, u^\ddagger) = (z^\star/z^\dagger, r^\star/r^\dagger, u^\star/u^\dagger)$ is a non-trivial solution to the SDP instance.

To see this, we first note that a given public key has exponentially many corresponding private keys and pk perfectly hides the vector $(\chi_1, \dots, \chi_n)$. Moreover, for a given pk, each message $(M_1, \dots, M_n)$ has an exponential number of valid signatures but the one produced by the signing algorithm is completely determined by $(\chi_1, \dots, \chi_n)$. We will see that, in $\mathcal{A}$'s view, guessing the value $z^\dagger$ of (1) amounts to inferring which vector $(\chi_1, \dots, \chi_n)$ the reduction $\mathcal{B}$ is using.

Throughout the game, $\mathcal{A}$ obtains signatures $\{(z_i, r_i, u_i)\}_{i=1}^{n-1}$ on at most $n-1$ linearly independent vectors of $\mathbb{G}^n$. If we consider discrete logarithms, these signatures only provide $\mathcal{A}$ with $n-1$ linearly independent equations because, for each triple (z_i, r_i, u_i), z_i uniquely determines (r_i, u_i). Taking into account the information revealed by $\{(g_i, h_i)\}_{i=1}^n$, we find that an unbounded adversary is presented with $3n-1$ linear equations in $3n$ unknowns. In $\mathcal{A}$'s view, since $(M_1^\star, \dots, M_n^\star)$ must be independent of previously signed vectors, predicting $z^\dagger$ is only possible with probability $1/p$. With probability $1-1/p$, we thus have $z^\dagger \neq z^\star$, in which case $(z^\ddagger, r^\ddagger, u^\ddagger)$ solves the SDP instance because $(z^\dagger, r^\dagger, u^\dagger)$ and $(z^\star, r^\star, u^\star)$ both satisfy the verification equations. □

The scheme can be modified so as to work in asymmetric pairing configurations and the Double Pairing assumption [1]. However, we need to work with the SDP assumption in the next section.

3.2 A Full-Fledged Linearly Homomorphic SPS Scheme

Here, we upgrade our one-time construction to obtain a scheme allowing us to sign an arbitrary number of linear subspaces. Here, each file identifier τ consists of a L-bit string. The construction builds on the observation that, in the scheme of Section 3.1, signatures (z, r, u) could be re-randomized by computing $(z \cdot g_r^\theta, r \cdot g_z^{-\theta}, u \cdot h_z^{-\log_h(g_r)\cdot\theta})$, with $\theta \xleftarrow{R} \mathbb{Z}_p$, if $h_z^{-\log_h(g_r)}$ were available. Since publicizing $h_z^{-\log_h(g_r)}$ would render the scheme insecure, our idea is to use Waters signatures as a support for introducing extra randomizers in the exponent.

In the construction, the u component of each signature can be seen as an aggregation of the one-time signature of Section 3.1 with a Waters signature $(h_z^{\log_h(g_r)} \cdot H_{\mathbb{G}}(\tau)^{-\rho}, h^\rho)$ [60] on the tag τ.

Keygen(λ, n): given a security parameter λ and the dimension $n \in \mathbb{N}$ of the subspace to be signed, choose bilinear group $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$. Then, conduct the following steps.

1. Choose $h \xleftarrow{R} \mathbb{G}$ and $\alpha_z, \alpha_r, \beta_z \xleftarrow{R} \mathbb{Z}_p$. Define $g_z = h^{\alpha_z}$, $g_r = h^{\alpha_r}$ and $h_z = h^{\beta_z}$.
2. For $i = 1$ to n, pick $\chi_i, \gamma_i, \delta_i \xleftarrow{R} \mathbb{Z}_p$ and compute $g_i = g_z^{\chi_i} g_r^{\gamma_i}$, $h_i = h_z^{\chi_i} h^{\delta_i}$.
3. Choose a random vector $\overline{\mathbf{w}} = (w_0, w_1, \dots, w_L) \xleftarrow{R} \mathbb{G}^{L+1}$. The latter defines a hash function $H_{\mathbb{G}} : \{0,1\}^L \to \mathbb{G}$ which maps $\tau = \tau[1] \dots \tau[L] \in \{0,1\}^L$ to $H_{\mathbb{G}}(\tau) = w_0 \cdot \prod_{k=1}^L w_k^{\tau[k]}$.

The private key is $\mathsf{sk} = \big(h_z^{\alpha_r},\ \{\chi_i, \gamma_i, \delta_i\}_{i=1}^n\big)$ while the public key consists of

$$\mathsf{pk} = \big(g_z,\ g_r,\ h_z,\ h,\ \{g_i, h_i\}_{i=1}^n,\ \overline{\mathbf{w}}\big) \in \mathbb{G}^{2n+4} \times \mathbb{G}^{L+1}.$$

Sign$(\mathsf{sk}, \tau, (M_1, \dots, M_n))$: to sign a vector $(M_1, \dots, M_n) \in \mathbb{G}^n$ w.r.t. the file identifier τ using $sk = \big(h_z^{\alpha_r},\ \{\chi_i, \gamma_i, \delta_i\}_{i=1}^n\big)$, choose $\theta, \rho \xleftarrow{R} \mathbb{Z}_p$ and output $\sigma = (z, r, u, v) \in \mathbb{G}^4$, where

$$z = g_r^\theta \cdot \prod_{i=1}^n M_i^{-\chi_i} \qquad\qquad r = g_z^{-\theta} \cdot \prod_{i=1}^n M_i^{-\gamma_i}$$

$$u = (h_z^{\alpha_r})^{-\theta} \cdot \prod_{i=1}^n M_i^{-\delta_i} \cdot H_{\mathbb{G}}(\tau)^{-\rho} \qquad\qquad v = h^\rho$$

$\mathsf{SignDerive}(\mathsf{pk}, \tau, \{(\omega_i, \sigma^{(i)})\}_{i=1}^{\ell})$: given pk, a file identifier τ and ℓ tuples $(\omega_i, \sigma^{(i)})$, parse $\sigma^{(i)}$ as $\sigma^{(i)} = \big(z_i, r_i, u_i, v_i\big) \in \mathbb{G}^4$ for $i = 1$ to ℓ. Then, choose $\rho' \overset{R}{\leftarrow} \mathbb{Z}_p$ and compute and return $\sigma = (z, r, u, v)$, where $z = \prod_{i=1}^{\ell} z_i^{\omega_i}$, $r = \prod_{i=1}^{\ell} r_i^{\omega_i}$, $u = \prod_{i=1}^{\ell} u_i^{\omega_i} \cdot H_{\mathbb{G}}(\tau)^{-\rho'}$ and $v = \prod_{i=1}^{\ell} v_i^{\omega_i} \cdot h^{\rho'}$.

$\mathsf{Verify}(\mathsf{pk}, \sigma, \tau, (M_1, \ldots, M_n))$: given a signature $\sigma = (z, r, u, v) \in \mathbb{G}^4$, a file identifier τ and a vector $(M_1, \ldots, M_n)$, return 1 if and only if $(M_1, \ldots, M_n) \neq (1_{\mathbb{G}}, \ldots, 1_{\mathbb{G}})$ and (z, r, u, v) satisfy

$$1_{\mathbb{G}_T} = e(g_z, z) \cdot e(g_r, r) \cdot \prod_{i=1}^{n} e(g_i, M_i), \qquad 1_{\mathbb{G}_T} = e(h_z, z) \cdot e(h, u) \cdot e(H_{\mathbb{G}}(\tau), v) \cdot \prod_{i=1}^{n} e(h_i, M_i). \qquad (2)$$

The security of the scheme against *non-independent* Type I adversaries is proved under the SDP assumption. In the case of Type II forgeries, we need to assume the adversary to be independent because, at some point, the simulator is only able to compute a signature for a unique value[5] of θ.

Theorem 2. *The scheme is unforgeable against independent adversaries if the SDP assumption holds in* $(\mathbb{G}, \mathbb{G}_T)$*. Moreover, the scheme is secure against non-independent Type I adversaries.*

Proof. The result is proved by separately considering Type I and Type II forgeries. For simplicity, we first consider Type II adversaries as the case of Type I attacks will be simpler. Lemmas 1 and 2 show how to build an algorithm solving the SDP problem either way. □

The proof of Lemma 1 uses Waters signatures as a handle to randomize signatures. Specifically, whenever the reduction is able to compute a Waters signatures $(h_z^{\alpha_r} \cdot H_{\mathbb{G}}(\tau)^{-\rho}, h^{\rho})$ on the tag τ, it can inject a fresh extra randomizer $\theta \in \mathbb{Z}_p$ in the exponent for each vector associated with τ. By doing so, with non-negligible probability, the specific vector $(\chi_1, \ldots, \chi_n)$ used by the reduction will remain completely undetermined from $\mathcal{A}$'s view.

Lemma 1. *For any Type II independent forger $\mathcal{A}$, there exists an algorithm $\mathcal{B}$ solving the SDP problem such that* $\mathbf{Adv}(\mathcal{A}) \leq 8 \cdot q \cdot (L+1) \cdot \big(\mathbf{Adv}^{\mathrm{SDP}}(\mathcal{B}) + \frac{1}{p}\big)$*, where q is the number of distinct tags appearing in signing queries.* (The proof is given in Appendix A.1).

Lemma 2. *A Type I forger $\mathcal{A}$ implies an algorithm $\mathcal{B}$ solving the SDP problem with non-negligible advantage. More precisely, we have* $\mathbf{Adv}(\mathcal{A}) \leq 8 \cdot q \cdot (L+1) \cdot \big(\mathbf{Adv}^{\mathrm{SDP}}(\mathcal{B}) + \frac{1}{p}\big)$*, where q is the number of distinct tags occurring in signing queries. Moreover, the statement holds even for non-independent adversaries.* (The proof is given in Appendix A.2).

Since the signature component u cannot be publicly randomized, the scheme does not have fully randomizable signatures. In Appendix B, we describe a fully randomizable variant. In applications like non-malleable commitments to group elements, the above scheme is sufficient however.

4 Applications

4.1 Verifiable Computation for Encrypted Cloud Storage

Linearly homomorphic schemes are known (see, *e.g.*, [8]) to provide verifiable computation mechanisms for outsourced data. Suppose that a user has a dataset consisting of n samples $s_1, \ldots, s_n \in \mathbb{Z}_p$. The dataset can be encoded as vectors $\vec{v}_i = (\vec{e}_i | s_i) \in \mathbb{Z}_p^{n+1}$, where $\vec{e}_i \in \mathbb{Z}_p^n$ denotes the i-th unit vector for each $i \in \{1, \ldots, n\}$. The user then assigns a file identifier τ to $\{\vec{v}_i\}_{i=1}^{n}$, computes signatures $\sigma_i \leftarrow \mathsf{Sign}(\mathsf{sk}, \tau, \vec{v}_i)$ on the resulting vectors and stores $\{(\vec{v}_i, \sigma_i)\}_{i=1}^{n}$ at the server. When requested, the server can then evaluate a sum $s = \sum_{i=1}^{n} s_i$ and provide evidence that the latter computation is correct by deriving a signature on the vector $(1, 1, \ldots, 1, s) \in \mathbb{Z}_p^{n+1}$. Unless the server is able to forge

[5] Note that this is not a problem since the signer can derive θ as a pseudorandom function of τ and $(M_1, \ldots, M_n)$ to make sure that a given vector is always signed using the same θ.

a signature for a vector outside the span of $\{\vec{v}_i\}_{i=1}^n$, it is unable to fool the user. The above method readily extends to authenticate weighted sums or Fourier transforms.

One disadvantage of the above method is that it requires the server to retain the dataset $\{s_i\}_{i=1}^n$ in the clear. Using linearly homomorphic structure-preserving signatures, the user can apply the above technique on encrypted samples using the Boneh-Boyen-Shacham (BBS) cryptosystem [16].

The BBS cryptosystem involves a public key $(g, \tilde{g}, f = g^x, h = g^y) \in_R \mathbb{G}^4$, where $(x, y) \in \mathbb{Z}_p^2$ is the private key. The user (or anyone else knowing his public key) can first encrypt his samples $\{s_i\}_{i=1}^n$ by computing BBS encryptions $(C_{1,i}, C_{2,i}, C_{3,i}) = (f^{r_i}, h^{t_i}, \tilde{g}^{s_i} \cdot g^{r_i+t_i})$, with $r_i, t_i \overset{R}{\leftarrow} \mathbb{Z}_p$, for each $i \in \{1, \ldots, n\}$. If the user holds a linearly homomorphic structure preserving signature key pair for vectors of dimension $n+3$, he can generate n structure preserving signatures on vectors $((C_{1,i}, C_{2,i}, C_{3,i})|\vec{E}_i) \in \mathbb{G}^{n+3}$, where $\vec{E}_i = (1_{\mathbb{G}}, \ldots, 1_{\mathbb{G}}, g, 1_{\mathbb{G}}, \ldots, 1_{\mathbb{G}}) = g^{\vec{e}_i}$ for each $i \in \{1, \ldots, n\}$, using the scheme of Section 3.2. The vectors $\{((C_{1,i}, C_{2,i}, C_{3,i})|\vec{E}_i)\}_{i=1}^n$ and their signatures $\{(z_i, r_i, u_i, v_i)\}_{i=1}^n$ are then archived in the cloud in such a way that the server can publicly derive a signature on the vector $\big(f^{\sum_i r_i}, h^{\sum_i t_i}, \tilde{g}^{\sum_i s_i} \cdot g^{\sum_i (r_i+t_i)}, g, g, \ldots, g\big) \in \mathbb{G}^{n+3}$ in order to convince the client that the encrypted sum was correctly computed. Using his private key (x, y), the client can then retrieve the sum $\sum_i s_i$ as long as it remains in a sufficiently small range.

The interest of the above solution lies in that the client can dispense with the need for storing the $O(n)$-size public key of his linearly homomorphic signature. Indeed, he can simply retain the random seed that was used to generate pk and re-compute private key elements $\{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^n$ whenever he wants to verify the server's response. In this case, the verification equations (2) become

$$1_{\mathbb{G}_T} = e(g_z, z \cdot \prod_{i=1}^n M_i^{\chi_i}) \cdot e(g_r, r \cdot \prod_{i=1}^n M_i^{\gamma_i}) = e(h_z, z \cdot \prod_{i=1}^n M_i^{\chi_i}) \cdot e(h, u \cdot \prod_{i=1}^n M_i^{\delta_i}) \cdot e(H_{\mathbb{G}}(\tau), v),$$

so that the client only has to compute $O(1)$ pairings. Moreover, the client does not have to determine an upper bound on the size of his dataset when generating his public key. Initially, he only needs to generate $\{(g_j, h_j)\}_{j=1}^3$. When the i-th ciphertext $(C_{1,i}, C_{2,i}, C_{3,i})$ has to be stored, the client derives $(\chi_{i+3}, \gamma_{i+3}, \delta_{i+3})$ and (g_{i+3}, h_{i+3}) by applying a PRF to the index i. This will be sufficient to sign vectors of the form $((C_{1,i}, C_{2,i}, C_{3,i})|\vec{E}_i)$.

In order to hide all partial information about the original dataset, the server may want to re-randomize the derived signature and ciphertext before returning them. This can be achieved by having the client include signatures on the vectors $(f, 1_{\mathbb{G}}, g, 1_{\mathbb{G}}, \ldots, 1_{\mathbb{G}})$, $(1_{\mathbb{G}}, h, g, 1_{\mathbb{G}}, \ldots, 1_{\mathbb{G}})$ in the outsourced dataset. Note that, in this case, the signature should be re-randomized as well. For this reason, our randomizable scheme described in Appendix B should be preferred.

Complete and careful security models for "verifiable computation on encrypted data" are beyond the scope of this paper. Here, they would naturally combine the properties of secure homomorphic encryption and authenticated computing. It should be intuitively clear that a malicious server cannot trick a client into accepting an incorrect result (*i.e.*, one which differs from the actual defined linear function it is supposed to compute over the defined signed ciphertext inputs) without defeating the security of the underlying homomorphic signature.

4.2 Extension to CCA1-Encrypted Data

In the application of Section 4.1, the underlying crypotosystem has to be additively homomorphic, which prevents it from being secure against adaptive chosen-ciphertext attacks. On the other hand, the method is compatible with security against *non-adaptive* chosen ciphertext attacks. One possibility is to apply the "lite" Cramer-Shoup technique (in its variant based on DLIN) as it achieves CCA1-security while remaining homomorphic. Unfortunately, the validity of ciphertexts is not publicly verifiable, which may be annoying in applications like cloud storage or universally verifiable e-voting systems. Indeed, servers may be willing to have guarantees that they are actually storing encryptions of some message instead of random group elements.

Consider the cryptosystem where ciphertexts $(C_1, C_2, C_3, C_4) = (f^r, h^t, g^{r+t}, \tilde{g}^m \cdot X_1^r \cdot X_2^t)$ are decrypted as $m = \log_{\tilde{g}}(C_4 \cdot C_1^{-x_1} C_2^{-x_2} C_3^{-z})$, where $X_1 = f^{x_1} g^z$ and $X_2 = h^{x_2} g^z$ are part of the public key. In [52], such a system was made chosen-ciphertext secure using a *publicly* verifiable one-time simulation-sound proof that (f, h, g, C_1, C_2, C_3) forms a DLIN tuple. In the security proof, if the reduction is guaranteed not to leak $C_1^{-x_1} C_2^{-x_2} C_3^{-z}$ for an invalid triple (C_1, C_2, C_3) (*i.e.*, as long as the adversary is unable to generate a fake proof for this), the private key component z will remain perfectly hidden. Consequently, if the challenge ciphertext is computed by choosing $C_3^\star \in_R \mathbb{G}$ (so that $(f, h, g, C_1^\star, C_2^\star, C_3^\star)$ is not a DLIN tuple) and computing $C_4^\star = \tilde{g}^m \cdot C_1^{\star x_1} \cdot C_2^{\star x_2} \cdot C_3^{\star z}$, the plaintext m is independent of $\mathcal{A}$'s view. If we replace the one-time simulation-sound proofs by standard proofs of membership in the scheme of [52], we obtain a CCA1 homomorphic encryption scheme. Linearly homomorphic SPS schemes provide a simple and efficient way to do that.

The idea is to include in the public key the verification key of a one-time linearly homomorphic SPS — using the scheme of Section 3.1— for $n = 3$ as well as signatures on the vectors $(f, 1_{\mathbb{G}}, g)$, $(1_{\mathbb{G}}, h, g) \in \mathbb{G}^3$. This will allow the sender to publicly derive a signature (z, r, u) on the vector $(C_1, C_2, C_3) = (f^r, h^t, g^{r+t})$. Each ciphertext thus consists of $(z, r, u, C_1, C_2, C_3, C_4)$. In the security proof, at each pre-challenge decryption query, the signature (z, r, u) serves as publicly verifiable evidence that (f, h, g, C_1, C_2, C_3) is a DLIN tuple. In the challenge phase, the reduction reveals another homomorphic signature $(z^\star, r^\star, u^\star)$ for a vector $(C_1^\star, C_2^\star, C_3^\star)$ that may be outside the span of $(f, 1_{\mathbb{G}}, g)$ and $(1_{\mathbb{G}}, h, g)$ but it does not matter since decryption queries are no longer allowed beyond this point.

We note that linearly homomorphic SPS can also be used to construct CCA1-secure homomorphic encryption schemes based on the Naor-Yung paradigm [56] in the standard model.

5 Non-Malleable Trapdoor Commitments to Group Elements from Linearly Homomorphic Structure-Preserving Signatures

As noted in [48, 49], some applications require to commit to group elements without knowing their discrete logarithms or destroying their algebraic structure by hashing them first. This section shows that, under a certain mild condition, linearly homomorphic SPS imply length-reducing non-malleable structure-preserving commitments to vectors of group elements.

As a result, we obtain the first length-reducing non-malleable structure-preserving trapdoor commitment. Our scheme is not *strictly*[6] structure-preserving (according to the terminology of [2]) because the commitment string lives in $\mathbb{G}_T$ rather than $\mathbb{G}$. Still, openings only consist of elements in $\mathbb{G}$, which makes it possible to generate efficient NIWI proofs that committed group elements satisfy certain properties. To our knowledge, the only known non-malleable commitment schemes whose openings only consist of group elements were described by Fischlin *et al.* [36]. However, these constructions cannot be length-reducing as they achieve universal composability [23, 25].

Our schemes are obtained by first constructing simulation-sound trapdoor commitments (SSTC) [40, 54] to group elements. SSTC schemes were first suggested by Garay, MacKenzie and Yang [40] as a tool for constructing universally composable zero-knowledge proofs [23]. MacKenzie and Yang subsequently gave a simplified security definition which suffices to provide non-malleability with respect to opening in the sense of the definition of re-usable non-malleable commitments [31].

In a SSTC, each commitment is labeled with a tag. The definition of [54] requires that, even if the adversary can see equivocations of commitments to possibly distinct messages for several tags $tag_1, \ldots, tag_q$, it will not be able to break the binding property for a new tag $tag \notin \{tag_1, \ldots, tag_q\}$.

[6] We recall that strictly structure-preserving commitments cannot be length-reducing, as shown by Abe *et al.* [2], so that our scheme is essentially the best we can hope for if we aim at short commitment stings.

Definition 5 ([54]). *A simulation-sound trapdoor commitment* $(\mathsf{Setup}, \mathsf{Com}, \mathsf{FakeCom}, \mathsf{FakeOpen}, \mathsf{Verify})$ *is a tuple where* $(\mathsf{Setup}, \mathsf{Com}, \mathsf{Verify})$ *forms a commitment scheme and* $(\mathsf{FakeCom}, \mathsf{FakeOpen})$ *are PPT algorithms with the following properties*

Trapdoor: *for any tag and any message* Msg*, the following distributions are computationally indistinguishable:*

$$D_{fake} := \{(pk, tk) \leftarrow \mathsf{Setup}(\lambda);\ (\widetilde{\mathsf{com}}, \mathsf{aux}) \leftarrow \mathsf{FakeCom}(pk, tk, tag);\\ \widetilde{\mathsf{dec}} \leftarrow \mathsf{FakeOpen}(\mathsf{aux}, tk, \widetilde{\mathsf{com}}, \mathsf{Msg}) : (pk, tag, \mathsf{Msg}, \widetilde{\mathsf{com}}, \widetilde{\mathsf{dec}})\}$$

$$D_{real} := \{(pk, tk) \leftarrow \mathsf{Setup}(\lambda);\ (\mathsf{com}, \mathsf{dec}) \leftarrow \mathsf{Com}(pk, tag, \mathsf{Msg}) : (pk, tag, \mathsf{Msg}, \mathsf{com}, \mathsf{dec})\}$$

Simulation-sound binding: *for any PPT adversary* $\mathcal{A}$*, the following probability is negligible*

$$\Pr[(pk, tk) \leftarrow \mathsf{Setup}(\lambda);\ (\mathsf{com}, tag, \mathsf{Msg}_1, \mathsf{Msg}_2, \mathsf{dec}_1, \mathsf{dec}_2) \leftarrow \mathcal{A}^{\mathcal{O}_{tk,pk}}(pk) : \mathsf{Msg}_1 \neq \mathsf{Msg}_2\\ \wedge \mathsf{Verify}(pk, tag, \mathsf{Msg}_1, \mathsf{com}, \mathsf{dec}_1) = \mathsf{Verify}(pk, tag, \mathsf{Msg}_2, \mathsf{com}, \mathsf{dec}_2) = 1 \wedge tag \notin Q],$$

where $\mathcal{O}_{tk,pk}$ *is an oracle that maintains an initially empty set* Q *and operates as follows:*

- *On input* (commit, tag)*, it runs* $(\widetilde{\mathsf{com}}, \mathsf{aux}) \leftarrow \mathsf{FakeCom}(pk, tk, tag)$*, stores* $(\widetilde{\mathsf{com}}, tag, \mathsf{aux})$*, returns* $\widetilde{\mathsf{com}}$ *and adds tag in* Q*.*
- *On input* $(\mathsf{decommit}, \widetilde{\mathsf{com}}, \mathsf{Msg})$*: if a tuple* $(\widetilde{\mathsf{com}}, tag, \mathsf{aux})$ *was previously stored, it computes* $\widetilde{\mathsf{dec}} \leftarrow \mathsf{FakeOpen}(\mathsf{aux}, tk, tag, \widetilde{\mathsf{com}}, \mathsf{Msg})$ *and returns* $\widetilde{\mathsf{dec}}$*. Otherwise,* $\mathcal{O}_{tk,pk}$ *returns* $\perp$*.*

While our SSTC to group elements will be proved secure in the above sense, a *non-adaptive* flavor of simulation-sound binding security is sufficient for the construction of non-malleable commitments. Indeed, Gennaro used [41] such a relaxed notion to achieve non-malleability from similar-looking multi-trapdoor commitments. In the non-adaptive notion, the adversary has to choose the set of tags $tag_1, \ldots, tag_\ell$ for which it wants to query the $\mathcal{O}_{tk,pk}$ oracle before seeing the public key pk.

5.1 Template of Linearly Homomorphic SPS Scheme

We first remark that *any* constant-size linearly homomorphic structure-preserving signature necessarily complies with the template below. Indeed, in order to have a linear homomorphism, each verification equation necessarily computes a product of pairings which should equal $1_{\mathbb{G}_T}$ in a valid signature. In each pairing of the product, one of the arguments must be a message or signature component while the second argument is either part of the public key or an encoding of the file identifier.

For simplicity, the template is described in terms of symmetric pairings but generalizations to asymmetric configurations are possible.

$\mathsf{\mathbf{Keygen}}(\boldsymbol{\lambda}, \boldsymbol{n})$**:** given λ and the dimension $n \in \mathbb{N}$ of the vectors to be signed, choose constants n_z, n_v, m. Among these, n_z and n_v will determine the signature length while m will be the number of verification equations. Then, choose $\{F_{j,\mu}\}_{j\in\{1,\ldots,m\},\mu\in\{1,\ldots,n_z\}}$, $\{G_{j,i}\}_{i\in\{1,\ldots,n\},\ j\in\{j,\ldots,m\}}$ in the group $\mathbb{G}$. The public key is $\mathsf{pk} = \big(\{F_{j,\mu}\}_{j\in\{1,\ldots,m\},\mu\in\{1,\ldots,n_z\}}, \{G_{j,i}\}_{i\in\{1,\ldots,n\},\ j\in\{j,\ldots,m\}}\big)$ while sk contains information about the representation of public elements w.r.t. specific bases.

$\mathsf{\mathbf{Sign}}(\mathsf{\mathbf{sk}}, \boldsymbol{\tau}, (\boldsymbol{M_1}, \ldots, \boldsymbol{M_n}))$**:** Outputs a tuple $\sigma = \big(Z_1, \ldots, Z_{n_z}, V_1, \ldots, V_{n_v}\big) \in \mathbb{G}^{n_z+n_v}$.

$\mathsf{\mathbf{SignDerive}}(\mathsf{\mathbf{pk}}, \boldsymbol{\tau}, \{(\boldsymbol{\omega_i}, \boldsymbol{\sigma^{(i)}})\}_{\boldsymbol{i=1}}^{\boldsymbol{\ell}})$**:** parses each $\sigma^{(i)}$ as $\big(Z_1^{(i)}, \ldots, Z_{n_z}^{(i)}, V_1^{(i)}, \ldots, V_{n_v}^{(i)}\big)$ and computes

$$Z_\mu = \prod_{i=1}^{\ell} Z_\mu^{(i)}\,{}^{\omega_i} \qquad V_\nu = \prod_{i=1}^{\ell} V_\nu^{(i)}\,{}^{\omega_i} \qquad \mu \in \{1, \ldots, n_z\},\ \nu \in \{1, \ldots, n_v\}.$$

After a possible extra re-randomization step, it outputs $\big(Z_1, \ldots, Z_{n_z}, V_1, \ldots, V_{n_v}\big)$.

Verify(pk, $\boldsymbol{\sigma}$, $\boldsymbol{\tau}$, $\boldsymbol{(M_1,\dots,M_n)}$): given a signature $\sigma = \big(Z_1,\dots,Z_{n_z},V_1,\dots,V_{n_v}\big) \in \mathbb{G}^{n_z+n_v}$, a tag τ and $(M_1,\dots,M_n)$, return 0 if $(M_1,\dots,M_n) = (1_{\mathbb{G}},\dots,1_{\mathbb{G}})$. Otherwise, do the following.

1. For each $j \in \{1,\dots,m\}$ and $\nu \in \{1,\dots,n_v\}$, compute one-to-one[7] encodings $T_{j,\nu} \in \mathbb{G}$ of the tag τ as a group element.
2. Return 1 if and only if $c_j = 1_{\mathbb{G}_T}$ for $j = 1$ to m, where

$$c_j = \prod_{\mu=1}^{n_z} e(F_{j,\mu}, Z_\mu) \cdot \prod_{\nu=1}^{n_v} e(T_{j,\nu}, V_\nu) \cdot \prod_{i=1}^{n} e(G_{j,i}, M_i) \qquad j \in \{1,\dots,m\}. \tag{3}$$

In the following, we say that a linearly homomorphic SPS is *regular* if, for each file identifier τ, any non-trivial vector $(M_1,\dots,M_n) \neq (1_{\mathbb{G}},\dots,1_{\mathbb{G}})$ has a valid signature.

5.2 Construction of Simulation-Sound Structure-Preserving Trapdoor Commitments

Let $\mathit{\Pi}^{\mathsf{SPS}} = (\mathsf{Keygen}, \mathsf{Sign}, \mathsf{SignDerive}, \mathsf{Verify})$ be a linearly homomorphic SPS. We construct a simulation-sound trapdoor commitment as follows.

SSTC.Setup($\boldsymbol{\lambda}$, $\boldsymbol{n}$): given the desired dimension $n \in \mathbb{N}$ of committed vectors, choose public parameters pp for the linearly homomorphic SPS scheme. Then, run $\mathit{\Pi}^{\mathsf{SPS}}.\mathsf{Keygen}(\lambda, n)$ to obtain a public key $\mathsf{pk} = \big(\{F_{j,\mu}\}_{j\in\{1,\dots,m\},\mu\in\{1,\dots,n_z\}}, \{G_{j,i}\}_{i\in\{1,\dots,n\},\ j\in\{j,\dots,m\}}\big)$, for some constants n_z, n_v, m, and a sk. The commitment key is $pk = \mathsf{pk}$ and the trapdoor tk consists of sk. Note that the public key defines a signature space $\mathbb{G}^{n_z+n_v}$, for constants n_z and n_v.

SSTC.Com($\boldsymbol{pk}$, $\boldsymbol{tag}$, $\boldsymbol{(M_1,\dots,M_n)}$): to commit to $(M_1,\dots,M_n) \in \mathbb{G}^n$ with respect to the tag $tag = \tau$, choose $\big(Z_1,\dots,Z_{n_z},V_1,\dots,V_{n_v}\big) \xleftarrow{R} \mathbb{G}^{n_z+n_v}$ in the signature space. Then, run step 1 of the verification algorithm and evaluate the right-hand-side member of (3). Namely, compute

$$c_j = \prod_{\mu=1}^{n_z} e(F_{j,\mu}, Z_\mu) \cdot \prod_{\nu=1}^{n_v} e(T_{j,\nu}, V_\nu) \cdot \prod_{i=1}^{n} e(G_{j,i}, M_i) \qquad j \in \{1,\dots,m\} \tag{4}$$

where $\{T_{j,\nu}\}_{j,\nu}$ form an injective encoding of $tag = \tau$ as a set of group elements. The commitment string is $\mathsf{com} = (c_1,\dots,c_m)$ whereas the decommitment is $\mathsf{dec} = \big(Z_1,\dots,Z_{n_z},V_1,\dots,V_{n_v}\big)$.

SSTC.FakeCom($\boldsymbol{pk}$, $\boldsymbol{tk}$, $\boldsymbol{tag}$): proceeds like $\mathsf{SSTC.Com}$ with $(\hat{M}_1,\dots,\hat{M}_n) \xleftarrow{R} \mathbb{G}^n$. If $(\hat{\mathsf{com}}, \hat{\mathsf{dec}})$ denotes the resulting pair, the algorithm outputs $\widetilde{\mathsf{com}} = \hat{\mathsf{com}}$ and the auxiliary information aux, which consists of the pair $\mathsf{aux} = ((\hat{M}_1,\dots,\hat{M}_n), \hat{\mathsf{dec}})$ for $tag = \tau$.

SSTC.FakeOpen(aux, $\boldsymbol{tk}$, $\boldsymbol{tag}$, $\widetilde{\mathbf{com}}$, $\boldsymbol{(M_1,\dots,M_n)}$): the algorithm parses $\widetilde{com}$ as $(\tilde{c}_1,\dots,\tilde{c}_m)$ and aux as $\big((\hat{M}_1,\dots,\hat{M}_n), (\hat{Z}_1,\dots,\hat{Z}_{n_z},\hat{V}_1,\dots,\hat{V}_{n_v})\big)$. It first generates a linearly homomorphic signature on $(M_1/\hat{M}_1,\dots,M_n/\hat{M}_n)$ for the tag $tag = \tau$. Namely, using the trapdoor $tk = \mathsf{sk}$, compute a signature $\sigma' = (Z'_1,\dots,Z'_{n_z},V'_1,\dots,V'_{n_v}) \leftarrow \mathit{\Pi}^{\mathsf{SPS}}.\mathsf{Sign}\big(\mathsf{sk}, \tau, (M_1/\hat{M}_n,\dots,M_n/\hat{M}_n)\big)$. Since σ' is a valid signature and $\mathsf{aux} = \big((\hat{M}_1,\dots,\hat{M}_n), (\hat{Z}_1,\dots,\hat{Z}_{n_z},\hat{V}_1,\dots,\hat{V}_{n_v})\big)$ satisfies

$$\tilde{c}_j = \prod_{\mu=1}^{n_z} e(F_{j,\mu}, \hat{Z}_\mu) \cdot \prod_{\nu=1}^{n_v} e(T_{j,\nu}, \hat{V}_\nu) \cdot \prod_{i=1}^{n} e(G_{j,i}, \hat{M}_i) \qquad j \in \{1,\dots,m\}, \tag{5}$$

the fake opening algorithm can run $(\tilde{Z}_1,\dots,\tilde{Z}_{n_z},\tilde{V}_1,\dots,\tilde{V}_{n_v}) \leftarrow \mathsf{SignDerive}(\mathsf{pk}, \tau, \{(1,\sigma'), (1,\hat{\sigma})\})$, where $\hat{\sigma} = (\hat{Z}_1,\dots,\hat{Z}_{n_z},\hat{V}_1,\dots,\hat{V}_{n_v})$, and output $\widetilde{\mathsf{dec}} = (\tilde{Z}_1,\dots,\tilde{Z}_{n_z},\tilde{V}_1,\dots,\tilde{V}_{n_v})$ which is a valid de-commitment to the vector $(M_1,\dots,M_n)$ with respect to $tag = \tau$.

[7] This condition can be relaxed to have collision-resistant deterministic encodings. Here, we assume injectivity for simplicity.

$\mathsf{SSTC.Verify}(pk, tag, (M_1, \ldots, M_n), \mathsf{com}, \mathsf{dec})$: parse com as $(c_1, \ldots, c_m) \in \mathbb{G}_T^m$ and the decommitment dec as $\big(Z_1, \ldots, Z_{n_z}, V_1, \ldots, V_{n_v}\big) \in \mathbb{G}^{n_z+n_v}$ (if these values do not parse properly, return 0). Then, compute a one-to-one encoding $\{T_{j,\nu}\}_{j,\nu}$ of $tag = \tau$. Return 1 if relations (4) hold and 0 otherwise.

In Appendix E, we extend the above construction so as to build simulation-sound trapdoor commitment to vectors from any linearly homomorphic signature that fits a certain template. As a result, we obtain a modular construction of constant-size non-malleable commitment to vectors which preserves the feasibility of efficiently proving properties about committed values.

Theorem 3. *Assuming that the underlying linearly homomorphic SPS is regular and secure against non-independent Type I adversaries, the above construction is a simulation-sound trapdoor commitment to group elements.* (The proof is given in Appendix D).

A standard technique (see, e.g., [40, 41]) to construct a re-usable non-malleable commitment from a SSTC scheme is as follows. To commit to Msg, the sender generates a key-pair $(\mathsf{VK}, \mathsf{SK})$ for a one-time signature and generates $(\mathsf{com}, \mathsf{dec}) \leftarrow \mathsf{SSTC.Commit}(pk, \mathsf{VK}, \mathsf{MSg})$ using VK as a tag. The non-malleable commitment string is the pair $(\mathsf{com}, \mathsf{VK})$ and the opening is given by (dec, σ), where σ is a one-time signature on com, so that the receiver additionally checks the validity of σ. This construction is known to provide independence (see Definition 8 in Appendix C) and thus non-malleability with respect to opening, as proved in [32, 44].

In our setting, we cannot compute σ as a signature of com, as it consists of $\mathbb{G}_T$ elements. However, we can rather sign the pair $(\mathsf{Msg}, \mathsf{dec})$ — whose components live in $\mathbb{G}$ — as long as it uniquely determines com. To this end, we can use the one-time structure-preserving of [1, Appendix C.1] since it allows signing messages of arbitrary length using a constant-size one-time public key. Like our scheme of Section 3.2, it relies on the SDP assumption and thus yields a non-malleable commitment based on this sole assumption. Alternatively, we can move σ in the commitment string (which thus consists of $(\mathsf{com}, \mathsf{VK}, \sigma)$), in which case the one-time signature does not need to be structure-preserving but it has to be strongly unforgeable (as can be observed from the definition of independent commitments [32] recalled in Appendix C) while the standard notion of unforgeability suffices in the former case.

Acknowledgements

The authors thank Dario Catalano for his comments and for pointing a necessary correction in the proof of Lemma 1.

References

1. M. Abe, K. Haralambiev, M. Ohkubo. Signing on Elements in Bilinear Groups for Modular Protocol Design. Cryptology ePrint Archive: Report 2010/133, 2010.
2. M. Abe, K. Haralambiev, M. Ohkubo. Group to Group Commitments Do Not Shrink. In *Eurocrypt'12*, *LNCS* 7237, pp. 301–317, 2012.
3. M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, M. Ohkubo. Structure-Preserving Signatures and Commitments to Group Elements. In *Crypto'10*, *LNCS* 6223, pp. 209–236, 2010.
4. M. Abe, J. Groth, K. Haralambiev, M. Ohkubo. Optimal Structure-Preserving Signatures in Asymmetric Bilinear Groups. In *Crypto'11*, *LNCS* 6841, pp. 649–666, 2011.
5. M. Abe, M. Chase, B. David, M. Kohlweiss, R. Nishimaki, M. Ohkubo. Constant-Size Structure-Preserving Signatures: Generic Constructions and Simple Assumptions. In *Asiacrypt'12*, *LNCS* 7658, pp. 4–24, 2012.
6. M. Abe, B. David, M. Kohlweiss, R. Nishimaki, M. Ohkubo. Tagged One-Time Signatures: Tight Security and Optimal Tag Size. In *PKC'13*, *LNCS* 7778, pp. 312–331, 2013.
7. M. Abe, J. Groth, M. Ohkubo. Separating Short Structure-Preserving Signatures from Non-interactive Assumptions. In *Asiacrypt'11*, *LNCS* 7073, pp. 628–646, 2011.
8. J.-H. Ahn, D. Boneh, J. Camenisch, S. Hohenberger, a. shelat, B. Waters. Computing on Authenticated Data. In *TCC 2012*, *LNCS* 7194, pp. 1–20, 2012.

9. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, D. Song. Provable data possession at untrusted stores. In *ACM-CCS 2007*, pp. 598–609, 2007.
10. G. Ateniese, S. Kamara, J. Katz. Proofs of Storage from Homomorphic Identification Protocols. In *Asiacrypt'09*, *LNCS* 5912, pp. 319–333, 2009.
11. N. Attrapadung, B. Libert. Homomorphic Network Coding Signatures in the Standard Model. In *PKC'11*, *LNCS* 6571, pp. 17–34, 2011.
12. N. Attrapadung, B. Libert, T. Peters. Computing on Authenticated Data: New Privacy Definitions and Constructions. In *Asiacrypt'12*, *LNCS* 7658, pp. 367–385, 2012.
13. N. Attrapadung, B. Libert, T. Peters. Efficient Completely Context-Hiding Quotable Signatures and Linearly Homomorphic Signatures. In *PKC'13*, *LNCS* 7778, pp. 367–385, pp. 386–404, 2013.
14. M. Bellare, T. Ristenpart. Simulation without the Artificial Abort: Simplified Proof and Improved Concrete Security for Waters' IBE Scheme. In *Eurocrypt'09*, *LNCS* 5479, pp. 407–424, 2009.
15. D. Boneh and X. Boyen. Short signatures without random oracles. In *Eurocrypt'04*, LNCS 3027, pages 56–73, 2004.
16. D. Boneh, X. Boyen, H. Shacham. Short Group Signatures. In *Crypto'04*, *LNCS* 3152, pp. 41–55. Springer, 2004.
17. D. Boneh, D. Freeman, J. Katz, B. Waters. Signing a Linear Subspace: Signature Schemes for Network Coding. In *PKC'09*, *LNCS* 5443, pp. 68–87, 2009.
18. D. Boneh, D. Freeman. Linearly Homomorphic Signatures over Binary Fields and New Tools for Lattice-Based Signatures. In *PKC'11*, *LNCS* 6571, pp. 1–16, 2011.
19. D. Boneh, D. Freeman. Homomorphic Signatures for Polynomial Functions. In *Eurocrypt'11*, *LNCS* 6632, pp. 149–168, 2011.
20. J. Camenisch, M. Dubovitskaya, K. Haralambiev. Efficient Structure-Preserving Signature Scheme from Standard Assumptions. In *Security and Cryptography for Networks 2012 (SCN 2012)*, *LNCS* 7485, pp. 76–94, 2012.
21. J. Camenisch, K. Haralambiev, M. Kohlweiss, J. Lapon, V. Naessens. Structure Preserving CCA Secure Encryption and Applications. In *Asiacrypt'11*, *LNCS* 7073, pp. 89–106, 2011.
22. J. Camenisch, T. Gross, T.-S. Heydt-Benjamin. Rethinking accountable privacy supporting services: extended abstract. In *Digital Identity Management 2008 (DIM'08)*, pp. 1–8, 2008.
23. R. Canetti. Universally Composable Security: A New Paradigm for Cryptographic Protocols. In *FOCS'01* pp. 136–145, 2001.
24. R. Canetti, Y. Dodis, R. Pass, S. Walfish. Universally Composable Security with Global Setup. In *TCC'07*, *LNCS* 4392, pp. 61–85, 2007.
25. R. Canetti, M. Fischlin. Universally Composable Commitments. In *Crypto'01*, *LNCS* 2139, pp. 19–40, 2001.
26. D. Catalano, D. Fiore, B. Warinschi. Adaptive Pseudo-free Groups and Applications. In *Eurocrypt'11*, *LNCS* 6632, pp. 207–223, 2011.
27. D. Catalano, D. Fiore, B. Warinschi. Efficient Network Coding Signatures in the Standard Model. In *PKC'12*, *LNCS* 7293, pp. 680–696, 2012.
28. J. Cathalo, B. Libert, M. Yung. Group Encryption: Non-Interactive Realization in the Standard Model. In *Asiacrypt'09*, *LNCS* 5912, pp. 179–196, 2009.
29. M. Chase, M. Kohlweiss. A New Hash-and-Sign Approach and Structure-Preserving Signatures from DLIN. In *Security and Cryptography for Networks 2012 (SCN 2012)*, *LNCS* 7485, pp. 131–148, 2012.
30. R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Crypto'98*, LNCS 1462, pages 13–25, 1998.
31. I. Damgård, J. Groth. Non-interactive and reusable non-malleable commitment schemes. In *STOC'03*, pages 426–437, 2003.
32. G. Di Crescenzo, Y. Ishai, R. Ostrovsky. Non-Interactive and Non-Malleable Commitment. In *STOC'98*, pp. 141–150, 1998.
33. Y. Desmedt. Computer security by redefining what a computer is. In *New Security Paradigms Workshop (NSPW) 1993*, pp. 160–166, 1993.
34. Y. Dodis, V. Shoup, S. Walfish. Efficient Constructions of Composable Commitments and Zero-Knowledge Proofs. In *Crypto'08*, *LNCS* 5157, pp. 21–38, 2008.
35. D. Dolev, C. Dwork, M. Naor. Non-malleable cryptography. In *STOC'91*, pages 542–552. ACM Press, 1991.
36. M. Fischlin, B. Libert, M. Manulis. Non-interactive and Re-usable Universally Composable String Commitments with Adaptive Security. In *Asiacrypt'11*, *LNCS* 7073, pp. 468–485, 2011.
37. D. Freeman. Improved security for linearly homomorphic signatures: A generic framework. In *PKC'12*, *LNCS* 7293, pp. 697–714, 2012.
38. G. Fuchsbauer. Automorphic Signatures in Bilinear Groups and an Application to Round-Optimal Blind Signatures. Cryptology ePrint Archive: Report 2009/320, 2009.
39. E. Fujisaki. New Constructions of Efficient Simulation-Sound Commitments Using Encryption and Their Applications. In *CT-RSA'12*, *LNCS* 7178, pp. 136–155, 2012.
40. J. Garay, P. MacKenzie, K. Yang Strengthening Zero-Knowledge Protocols Using Signatures. In *Eurocrypt'03*, *LNCS* 2656, pp. 177–194, 2003.

41. R. Gennaro. Multi-trapdoor Commitments and Their Applications to Proofs of Knowledge Secure Under Concurrent Man-in-the-Middle Attacks. In *Crypto'04*, *LNCS* 3152, pp. 220–236, 2004.
42. R. Gennaro, C. Gentry, B. Parno. Non-interactive Verifiable Computing: Outsourcing Computation to Untrusted Workers. In *Crypto 2010*, *LNCS* 6223, pp. 465–482, 2010.
43. R. Gennaro, J. Katz, H. Krawczyk, T. Rabin. Secure Network Coding over the Integers. In *PKC'10*, *LNCS* 6056, pp. 142–160, 2010.
44. R. Gennaro and S. Micali. Independent Zero-Knowledge Sets. In *ICALP'06*, LNCS 4052, pages 34–45, 2006.
45. J. Groth. Simulation-Sound NIZK Proofs for a Practical Language and Constant Size Group Signatures. In *Asiacrypt'06*, *LNCS* 4284, pp. 444–459, Springer, 2006.
46. J. Groth, R. Ostrovsky. Cryptography in the Multi-String Model. In *Crypto'07*, LNCS 4622, pp. 323–341, 2007.
47. J. Groth, A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Eurocrypt'08*, LNCS 4965, pp. 415–432, 2008.
48. J. Groth. Homomorphic trapdoor commitments to group elements. Cryptology ePrint Archive: Report 2009/007, 2009.
49. J. Groth. Efficient Zero-Knowledge Arguments from Two-Tiered Homomorphic Commitments. In *Asiacrypt'11*, *LNCS* 7073, pp. 431–448, 2011.
50. D. Hofheinz, E. Kiltz. Programmable Hash Functions and Their Applications. In *Crypto'08*, *LNCS* 5157, pp. 21–38, 2008.
51. D. Hofheinz, T. Jager. Tightly Secure Signatures and Public-Key Encryption. In *Crypto'12*, *LNCS* 7417, pp. 590–607, 2012.
52. B. Libert, M. Yung. Non-Interactive CCA2-Secure Threshold Cryptosystems with Adaptive Security: New Framework and Constructions. In *TCC 2012*, LNCS 7194, pp. 75–93, Springer, 2012.
53. R. Johnson, D. Molnar, D. Song, D. Wagner. Homomorphic Signature Schemes. In *CT-RSA'02*, LNCS 2271, pp. 244–262, 2002.
54. P. MacKenzie, K. Yang. On Simulation-Sound Trapdoor Commitments. In *Eurocrypt'04*, *LNCS* 3027, pp. 382–400, 2004.
55. T. Malkin, I. Teranishi, Y. Vahlis, M. Yung. Signatures resilient to continual leakage on memory and computation. In *TCC'11*, 89–106, 2011.
56. M. Naor, M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC'90*, ACM Press, 1990.
57. R. Nishimaki, E. Fujisaki, K. Tanaka. A Multi-trapdoor Commitment Scheme from the RSA Assumption. In *ACISP 2010*, *LNCS*, 6168, pp. 182-199, 2010.
58. Y. Sakai, K. Emura, G. Hanaoka, Y. Kawai, T. Matsuda, and K. Omote. Group Signatures with Message-Dependent Opening. In *5th International Conference on Pairing-Based Cryptography (Pairing 2012)*, LNCS 7708, pp. 270–294, 2013.
59. A. Shamir. Identity-Based Cryptosystems and Signature Schemes. In *Crypto'84*, *LNCS* 196, pp. 47–53, 1984.
60. B. Waters. Efficient Identity-Based Encryption Without Random Oracles. In *Eurocrypt'05*, *LNCS* 3494, pp. 114–127, 2005.

A Deferred Proofs for the Scheme in Section 3.2

A.1 Proof of Lemma 1

Proof. Let us assume that an *independent* adversary $\mathcal{A}$ can produce a Type II forgery with non-negligible advantage ε. Using $\mathcal{A}$, we build an algorithm $\mathcal{B}$ solving a SDP instance (g_z, g_r, h_z, h) with probability at least $\varepsilon/(8(q-1)(L+1))$. Algorithm $\mathcal{B}$ chooses $(w_0, w_1, \ldots, w_L) \in \mathbb{G}^{L+1}$ in the same way as in the security proof of Waters signatures [60]. Namely, for any string $\tau \in \{0,1\}^L$, the hash value $H_{\mathbb{G}}(\tau) = w_0 \cdot \prod_{i=1}^{L} w_i^{\tau[i]}$ can be expressed as $H_{\mathbb{G}}(\tau) = g_r^{J(\tau)} \cdot h^{K(\tau)}$ for certain integer-valued functions $J, K : \{0,1\}^L \to \mathbb{Z}_p$ that remain internal to the simulation. They are further defined using the methodology of programmable hash functions [50] so that, for any distinct $\tau, \tau_1, \ldots, \tau_q$, we have $J(\tau) = 0 \bmod p$ and $J(\tau_i) \neq 0 \bmod p$ for each $i \in \{1, \ldots, q\}$ with non-negligible probability $\zeta = 1/(8 \cdot q \cdot (L+1))$.

Remaining public key components are defined by setting $g_i = g_z^{\chi_i} g_r^{\gamma_i}$ and $h_i = h_z^{\chi_i} h^{\delta_i}$, with $\chi_i, \gamma_i, \delta_i \xleftarrow{R} \mathbb{Z}_p$ for $i = 1$ to n, as in the real key generation algorithm.

Since $\mathcal{A}$ is a Type II forger, it is expected to produce a forgery $(\tau^\star, \vec{M}^\star, \sigma^\star)$ for a tag $\tau^\star$ that was used by $\mathcal{B}$ in some signing query but for which $\vec{M}^\star \notin \mathrm{span}(\vec{M}_1, \ldots, \vec{M}_{n-1})$, where $\vec{M}_1, \ldots, \vec{M}_{n-1}$ are the vectors of $\mathbb{G}^n$ that were associated with $\tau^\star$. We denote by $\tau_1, \ldots, \tau_q$ the distinct adversarially-chosen tags involved in $\mathcal{A}$'s queries during the game. Note that, since $\mathcal{A}$ is a Type II adversary, we will have $\tau^\star \in \{\tau_1, \ldots, \tau_q\}$ at the end of the game. We also assume w.l.o.g. that exactly $n-1$ signing queries are made for each tag $\tau \in \{\tau_1, \ldots, \tau_q\}$ during the game (otherwise, $\mathcal{B}$ can simulate signing queries for itself). During its interaction with $\mathcal{A}$, the reduction $\mathcal{B}$ answers Sign, $\mathsf{SignDerive}$ and Reveal queries as follows.

Signing queries: At each signing query $\big(\tau_j, \vec{M} = (M_1, \ldots, M_n)\big)$ involving the j-th distinct tag τ_j, $\mathcal{B}$ evaluates the function $J(\tau_j)$ and considers the following situations.

– If $J(\tau_j) \neq 0$, $\mathcal{B}$ picks $\rho, \theta \xleftarrow{R} \mathbb{Z}_p$ and computes

$$\Theta_1 = H_{\mathbb{G}}(\tau_j)^{-\rho} \cdot (h_z)^{\frac{K(\tau_j)}{J(\tau_j)} \cdot \theta} \cdot, \qquad\qquad \Theta_2 = h^\rho \cdot (h_z)^{\frac{-\theta}{J(\tau_j)}},$$

which can be written $(\Theta_1, \Theta_2) = \big(h_z^{-\theta \cdot \alpha_r} \cdot H_{\mathbb{G}}(\tau)^{-\tilde{\rho}}, h^{\tilde{\rho}}\big)$ if we define $\tilde{\rho} = \rho - \frac{\theta \cdot \beta_z}{J(\tau_j)}$. Using (Θ_1, Θ_2), $\mathcal{B}$ obtains a valid signature on the vector $(M_1, \ldots, M_n)$ by computing

$$z = g_r^\theta \cdot \prod_{i=1}^n M_i^{-\chi_i} \qquad r = g_z^{-\theta} \cdot \prod_{i=1}^n M_i^{-\gamma_i} \qquad u = \Theta_1 \cdot \prod_{i=1}^n M_i^{-\delta_i} \qquad v = \Theta_2$$

The signature $\sigma = (z, r, u, v)$ is not directly sent to $\mathcal{A}$ but assigned to a new handle h and stored in an entry $(\mathsf{h}, (\tau_j, \vec{M}), \sigma)$ of the table T.

– If $J(\tau_j) = 0$, $\mathcal{B}$ picks $\rho \xleftarrow{R} \mathbb{Z}_p$ and computes

$$z = \prod_{i=1}^n M_i^{-\chi_i} \qquad r = \prod_{i=1}^n M_i^{-\gamma_i} \qquad u = H_{\mathbb{G}}(\tau_j)^{-\rho} \cdot \prod_{i=1}^n M_i^{-\delta_i} \qquad v = h^\rho,$$

which corresponds to a valid signature (z, r, u, v) on $(M_1, \ldots, M_n)$ for which $\theta = 1$. Again, $\mathcal{B}$ chooses a handle h and stores $\big(\mathsf{h}, (\tau, \vec{M}), (z, r, u, v)\big)$ in the table T.

Derivation queries: Whenever $\mathcal{A}$ queries $\big((\mathsf{h}_1, \ldots, \mathsf{h}_k), (\tau, \vec{M}'), \{\beta_i\}_{i=1}^k\big)$ to the $\mathsf{SignDerive}$ oracle, $\mathcal{B}$ returns $\perp$ if not all handles $\mathsf{h}_1, \ldots, \mathsf{h}_k$ correspond to queries involving τ. Otherwise, let $\vec{M}_1, \ldots, \vec{M}_k$ be the queried vectors. If $\vec{M}' \neq \prod_{i=1}^k \vec{M}_i^{\beta_i}$, $\mathcal{B}$ returns $\perp$. Otherwise, $\mathcal{B}$ answers the query in the same way as the real $\mathsf{SignDerive}$ oracle, by updating the table T.

Reveal queries: When $\mathcal{A}$ supplies a handle h, $\mathcal{B}$ returns $\perp$ if no entry of the form $(\mathsf{h}, (\tau, \vec{M}), .)$ exists in T. Otherwise, $\mathcal{B}$ returns the previously generated signature σ and adds $\big((\tau, \vec{M}), \sigma\big)$ in the list Q.

Forgery: Eventually, $\mathcal{A}$ outputs a Type II forgery $(\tau^\star, \vec{M}^\star, \sigma^\star)$, where $\vec{M}^\star = (M_1^\star, \ldots, M_n^\star)$ and $\sigma^\star = (z^\star, r^\star, u^\star, v^\star) \in \mathbb{G}^4$ satisfies the verification equation. At this point, $\mathcal{B}$ evaluates $J(\tau^\star)$ and reports failure if $J(\tau^\star) \neq 0$ or if the set $\{\tau_1, \ldots, \tau_q\}$ contains at least two tags τ_{j_1}, τ_{j_2} such that $J(\tau_{j_1}) = J(\tau_{j_2}) = 0$. The same analysis as in [60] shows that, with probability $1/(8(q-1)(L+1))$, we have $J(\tau^\star) = 0$ and $J(\tau_j) \neq 0$ for each $\tau_j \in \{\tau_1, \ldots, \tau_q\} \backslash \{\tau^\star\}$. We thus find that $\mathcal{B}$'s probability not to abort during the entire game is at least $1/(8(q-1)(L+1))$.

If $\mathcal{B}$ does not fail, we have $H_{\mathbb{G}}(\tau^\star) = h^{K(\tau^\star)}$, so that $\mathcal{B}$ can compute

$$z^\dagger = \prod_{i=1}^n M_i^{\star - \chi_i} \qquad r^\dagger = \prod_{i=1}^n M_i^{\star - \gamma_i} \qquad u^\dagger = v^{\star - K(\tau^\star)} \cdot \prod_{i=1}^n M_i^{\star - \delta_i} \qquad v^\dagger = v^\star. \tag{6}$$

We see that $(z^\dagger, r^\dagger, u^\dagger, v^\dagger)$ forms a valid signature on $(M_1^\star, \ldots, M_n^\star)$ whose last component $v^\dagger$ coincides with that of $\mathcal{A}$'s forgery. Since $(z^\dagger, r^\dagger, u^\dagger, v^\dagger)$ and $(z^\star, r^\star, u^\star, v^\star)$ both satisfy the verification equations, the triple

$$(z^\ddagger, r^\ddagger, u^\ddagger) = \Big(\frac{z^\star}{z^\dagger}, \frac{r^\star}{r^\dagger}, \frac{u^\star}{u^\dagger}\Big)$$

necessarily satisfies $e(g_z, z^{\ddagger}) \cdot e(g_r, r^{\ddagger}) = e(h_z, z^{\ddagger}) \cdot e(h, u^{\ddagger}) = 1_{\mathbb{G}_T}$. We are thus left with proving that $z^{\ddagger} \neq 1_{\mathbb{G}}$ with all but negligible probability.

To do this, the key observation is that, in the desirable event

$$J(\tau^{\star}) = 0 \qquad \wedge \qquad \bigwedge_{\tau_j \neq \tau^{\star}} J(\tau_j) \neq 0, \tag{7}$$

the only information that $\mathcal{B}$ reveals about $(\chi_1, \ldots, \chi_n)$ is contained in the z-components of signatures involving $\tau^{\star}$ if $\mathcal{A}$ is a Type II adversary. Indeed, for each signing query $(\tau, \vec{M})$ such that $\tau \neq \tau^{\star}$, $\mathcal{B}$ introduces in the signature a fresh random exponent $\theta \in_R \mathbb{Z}_p$ that does not appear anywhere else. This allows $\mathcal{B}$ not to leak anything about $(\chi_1, \ldots, \chi_n)$ during these queries.

More precisely, let us first consider what an unbounded Type II adversary $\mathcal{A}$ can see. Throughout the game, $\mathcal{A}$ makes $n(q-1) + (n-1)$ signing queries since at most $n-1$ independent queries are allowed for the tag $\tau^{\star}$. Let us index these queries as $\{(\tau_j, \vec{M}_k = (M_{k,1}, \ldots, M_{k,n}))\}_{j,k}$, with $j \in \{1, \ldots, q\}$, and let $\{(z_{j,k}, r_{j,k}, u_{j,k}, v_{j,k})\}_{j,k}$ denote the answers in which $\mathcal{B}$ introduces $n(q-1)$ variables $\{\theta_{j,k}\}_{j \neq j^{\star}, k \in \{1,\ldots,n\}}$ in the exponent. Together with private key elements $\{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^{n}$, we have a total of $3n + n(q-1) = 2n + nq$ unknowns. Each signature $(z_{j,k}, r_{j,k}, u_{j,k}, v_{j,k})$ provides $\mathcal{A}$ with at most one new linearly independent equation —recall that $(z_{j,k}, v_{j,k})$ uniquely determines $r_{j,k}, u_{j,k}$ while $v_{j,k}$ does not depend on $\theta_{j,k}$ or $\{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^{n}$—in addition to the $2n$ linear equations resulting from the public key elements $\{(g_i, h_i)\}_{i=1}^{n}$.

Overall, a Type II adversary $\mathcal{A}$ thus obtains $2n + nq - 1$ linear equations which is insufficient to solve a system of $2n + nq$ unknowns. Since $(M_1^{\star}, \ldots, M_n^{\star})$ is linearly independent of the vectors $\vec{M}_{j^{\star},1}, \ldots, \vec{M}_{j^{\star},n-1}$ associated with $\tau^{\star}$, for $\mathcal{A}$, predicting the value $z^{\ddagger}$ of (8) is equivalent to finding the missing piece equation that would determine $(\chi_1, \ldots, \chi_n)$. With probability $1 - 1/p$, we thus have $z^{\ddagger} \neq z^{\star}$ as claimed. □

A.2 Proof of Lemma 2

Proof. Let $\mathcal{A}$ be a Type I forger with non-negligible advantage ε. We show that it implies an algorithm $\mathcal{B}$ solving a SDP instance (g_z, g_r, h_z, h) with probability at least $\varepsilon/(8q(L+1))$.

Algorithm $\mathcal{B}$ begins by choosing $(w_0, w_1, \ldots, w_L) \in \mathbb{G}^{L+1}$ as in the security proof of Waters signatures [60]. This is done in such a way that, for any $\tau \in \{0,1\}^L$, the hash value $H_{\mathbb{G}}(\tau)$ can be written $H_{\mathbb{G}}(\tau) = g_r^{J(\tau)} \cdot h^{K(\tau)}$ for the same functions $J, K : \{0,1\}^L \to \mathbb{Z}_p$ as in the proof of Lemma 1. For any distinct $\tau, \tau_1, \ldots, \tau_q$, we will thus have $J(\tau) = 0 \bmod p$ and $J(\tau_i) \neq 0 \bmod p$ for each $i \in \{1, \ldots, q\}$ with non-negligible probability $\zeta = 1/(8 \cdot q \cdot (L+1))$.

Other public key components are defined by setting $g_i = g_z^{\chi_i} g_r^{\gamma_i}$ and $h_i = h_z^{\chi_i} h^{\delta_i}$, with $\chi_i, \gamma_i, \delta_i \xleftarrow{R} \mathbb{Z}_p$ for $i = 1$ to n. During the game, $\mathcal{A}$'s queries are handled as follows.

Signing queries: At each signing query $(\tau_j, \vec{M} = (M_1, \ldots, M_n))$ involving the j-th distinct tag τ_j, $\mathcal{B}$ aborts in the event that $J(\tau_j) = 0 \bmod p$. Otherwise, $\mathcal{B}$ picks $\rho, \theta \xleftarrow{R} \mathbb{Z}_p$ and computes

$$\varTheta_1 = H_{\mathbb{G}}(\tau_j)^{-\rho} \cdot (h_z)^{\frac{K(\tau_j)}{J(\tau_j)} \cdot \theta} \cdot, \qquad \varTheta_2 = h^{\rho} \cdot (h_z)^{\frac{-\theta}{J(\tau_j)}}.$$

Note that the above pair can be written $(\varTheta_1, \varTheta_2) = (h_z^{-\theta \cdot \alpha_r} \cdot H_{\mathbb{G}}(\tau)^{-\tilde{\rho}}, h^{\tilde{\rho}})$, where $\tilde{\rho} = \rho - \frac{\theta \cdot \beta_z}{J(\tau_j)}$. Using $(\varTheta_1, \varTheta_2)$, $\mathcal{B}$ obtains a well-formed signature on $(M_1, \ldots, M_n)$ by computing

$$z = g_r^{\theta} \cdot \prod_{i=1}^{n} M_i^{-\chi_i} \qquad r = g_z^{-\theta} \cdot \prod_{i=1}^{n} M_i^{-\gamma_i} \qquad u = \varTheta_1 \cdot \prod_{i=1}^{n} M_i^{-\delta_i} \qquad v = \varTheta_2.$$

The signature $\sigma = (z, r, u, v)$ is not directly returned to $\mathcal{A}$ but associated with a new handle h and stored in an entry $(\mathsf{h}, (\tau_j, \vec{M}), \sigma)$ of the table T.

Derivation queries: When $\mathcal{A}$ queries $\big((\mathsf{h}_1,\ldots,\mathsf{h}_k),(\tau,\vec{M}'),\{\beta_i\}_{i=1}^k\big)$ to the signature derivation oracle, $\mathcal{B}$ returns $\perp$ if not all handles $\mathsf{h}_1,\ldots,\mathsf{h}_k$ correspond to queries involving τ. Otherwise, let $\vec{M}_1,\ldots,\vec{M}_k$ be the queried vectors. If $\vec{M}' \neq \prod_{i=1}^k \vec{M}_i^{\beta_i}$, $\mathcal{B}$ returns $\perp$. Otherwise, $\mathcal{B}$ answers exactly like the real $\mathsf{SignDerive}$ oracle and updates the table T.

Reveal queries: When $\mathcal{A}$ queries the Reveal oracle with a handle h, $\mathcal{B}$ returns $\perp$ if no entry of the form $(\mathsf{h},(\tau,\vec{M}),.)$ exists in T. Otherwise, $\mathcal{B}$ returns the previously computed signature σ —just like the actual Reveal oracle— and adds $\big((\tau,\vec{M}),\sigma\big)$ in the list Q.

Forgery: Eventually, $\mathcal{A}$ outputs a Type II forgery $(\tau^\star,\vec{M}^\star,\sigma^\star)$, where $\vec{M}^\star=(M_1^\star,\ldots,M_n^\star)$ and $\sigma^\star=(z^\star,r^\star,u^\star,v^\star)\in\mathbb{G}^4$ is a tuple satisfying the verification equation. At this step, $\mathcal{A}$ computes $J(\tau^\star)$ and aborts if $J(\tau^\star)\neq 0$. However, the same analysis as in [60] shows that, with probability $1/(8q(L+1))$, we have $J(\tau^\star)=0$ and $J(\tau_j)\neq 0$ for each $j\in\{1,\ldots,q\}$.

If $\mathcal{B}$ does not fail, we have $H_{\mathbb{G}}(\tau^\star)=h^{K(\tau^\star)}$ and $\mathcal{B}$ can thus compute

$$z^\dagger=\prod_{i=1}^n M_i^{\star-\chi_i} \qquad r^\dagger=\prod_{i=1}^n M_i^{\star-\gamma_i} \qquad u^\dagger=v^{\star-K(\tau^\star)}\cdot\prod_{i=1}^n M_i^{\star-\delta_i} \qquad v^\dagger=v^\star. \tag{8}$$

The 4-uple $(z^\dagger,r^\dagger,u^\dagger,v^\dagger)$ forms a valid signature on $(M_1^\star,\ldots,M_n^\star)$ whose last component is identical to that of $\mathcal{A}$'s forgery. Since $(z^\dagger,r^\dagger,u^\dagger,v^\dagger)$ and $(z^\star,r^\star,u^\star,v^\star)$ both satisfy the verification equations, we find that

$$(z^\ddagger,r^\ddagger,u^\ddagger)=\Big(\frac{z^\star}{z^\dagger},\frac{r^\star}{r^\dagger},\frac{u^\star}{u^\dagger}\Big)$$

necessarily gives a non-trivial solution to the SDP instance with overwhelming probability.

Indeed, the same arguments as in the proof of Lemma 1 show that we can only have $z^\ddagger\neq 1_{\mathbb{G}}$ with probability $1/p$. The reason is that, in each signing query, $\mathcal{B}$ introduces a new blinding exponent θ that does not appear anywhere else. For this reason, $\mathcal{B}$ never leaks any information about $(\chi_1,\ldots,\chi_n)$ at any time and the element $z^\dagger$ is thus completely undetermined in $\mathcal{A}$'s view. □

B A Fully Randomizable Linearly Homomorphic SPS

In certain situations, one may want derived signatures to have the same distribution as original signatures on the same messages.

B.1 Privacy Definition

Ahn *et al.* [8] formalized a strong privacy property requiring that derived signatures be statistically indistinguishable from original ones, even when these are given.

In [12], Attrapadung *et al.* extended the definition of [8] —which only considers honestly generated signatures— to any original signature satisfying the verification algorithm.

Definition 6 ([12]). *A linearly homomorphic signature* $(\mathsf{Keygen},\mathsf{Sign},\mathsf{SignDerive},\mathsf{Verify})$ *is said* completely context hiding *if, for all public/private key pairs* $(\mathsf{pk},\mathsf{sk})\leftarrow\mathsf{Keygen}(\lambda)$*, for any message set* $\mathcal{S}=\{(\tau,\vec{M}_1),\ldots,(\tau,\vec{M}_{n-1})\}$*, any coefficients* $\{\omega_i\}_{i=1}^{n-1}$ *and any* $(\tau,\vec{M})$ *such that* $\vec{M}=\prod_{i=1}^{n-1}\vec{M}_i^{\omega_i}$*, for all* $\{\sigma_i\}_{i=1}^{n-1}$ *such that* $\mathsf{Verify}(\mathsf{pk},\tau,\vec{M}_i,\sigma_i)=1$*, the following distributions are statistically close*

$$\Big\{\big(\mathsf{sk},\ \{\sigma_i\}_{i=1}^{n-1},\ \mathsf{Sign}(\mathsf{sk},\tau,\vec{M})\big)\Big\}_{\mathsf{sk},\mathcal{S},\vec{M}}, \qquad \big\{\big(\mathsf{sk},\ \{\sigma_i\}_{i=1}^{n-1},\ \mathsf{SignDerive}\big(\mathsf{pk},\tau,\ \{(\omega_i,\sigma_i)\}_{i=1}^{n-1}\big)\big)\big\}_{\mathsf{sk},\mathcal{S},\vec{M}}.$$

In [8] Ahn *et al.* showed that, if a scheme is strongly context hiding, then Definition 1 can be simplified by removing the $\mathsf{SignDerive}$ and Reveal oracles and only providing the adversary with an ordinary signing oracle.

B.2 A Completely Context-Hiding Construction

We show that our scheme of Section 3.2 can be modified so as to become *strongly* context-hiding in the sense of [8]. Namely, signatures produced by the $\mathsf{SignDerive}$ algorithm should be statistically indistinguishable from signatures freshly generated by Sign, even when the original signatures are given.

The difficulty is that, in the scheme of Section 3.2, we cannot re-randomize the underlying θ without knowing $h_z^{\alpha_r}$. To address this problem, it is tempting to include in each signature a randomization component of the form $(h_z^{\alpha_r} \cdot H_{\mathbb{G}}(\tau)^{-\zeta}, h^{\zeta})$, for some $\zeta \in \mathbb{Z}_p$, which can be seen as a signature on the vector $(1_{\mathbb{G}}, \ldots, 1_{\mathbb{G}})$. Unfortunately, the security proof ceases to go through as the reduction finds itself unable to generate a well-formed pair $(h_z^{\alpha_r} \cdot H_{\mathbb{G}}(\tau)^{-\zeta}, h^{\zeta})$ at some step of its interaction with the adversary. Our solution actually consists in committing to the signature components that cannot be re-randomized and provide evidence that committed group elements satisfy the verification equations. This is achieved using Groth-Sahai non-interactive arguments on a perfectly witness indistinguishable Groth-Sahai CRS, as in the linearly homomorphic construction of Attrapadung *et al.* [13]. A slight difference with [13], however, is that signature components $(H_{\mathbb{G}}(\tau)^{-\rho}, h^{-\rho})$ are no longer used and replaced by the technique of Malkin *et al.* [55], which yields slightly shorter signatures.

$\mathsf{Keygen}(\boldsymbol{\lambda}, \boldsymbol{n})$: given a security parameter λ and the dimension $n \in \mathbb{N}$ of the subspace to be signed, choose bilinear group $(\mathbb{G}, \mathbb{G}_T)$ of order $p > 2^{\lambda}$. Then, do the following.

1. Choose $h \xleftarrow{R} \mathbb{G}$ and $\alpha_z, \alpha_r, \beta_z, \xleftarrow{R} \mathbb{Z}_p$. Define $g_z = h^{\alpha_z}$, $g_r = h^{\alpha_r}$ and $h_z = h^{\beta_z}$.
2. For each $i \in \{1, \ldots, n\}$, pick $\chi_i, \gamma_i, \delta_i \xleftarrow{R} \mathbb{Z}_p$ and compute $g_i = g_z^{\chi_i} \cdot g_r^{\gamma_i}$, $h_i = h_z^{\chi_i} \cdot h^{\delta_i}$.
3. Generate $L+1$ Groth-Sahai common reference strings by choosing $f_1, f_2 \xleftarrow{R} \mathbb{G}$ and defining vectors $\vec{f_1} = (f_1, 1, g) \in \mathbb{G}^3$, $\vec{f_2} = (1, f_2, g) \in \mathbb{G}^3$ and $\vec{f}_{3,i} \xleftarrow{R} \mathbb{G}^3$, for each $i \in \{0, \ldots, L\}$.

The public key consists of

$$\mathsf{pk} = \Big(g_z,\ g_r,\ h_z,\ h,\ \{g_i, h_i\}_{i=1}^{n},\ \mathbf{f} = \big(\vec{f_1}, \vec{f_2}, \{\vec{f}_{3,i}\}_{i=0}^{L}\big) \Big)$$

while the private key is $\mathsf{sk} = \big(h_z^{\alpha_r},\ \{\chi_i, \gamma_i, \delta_i\}_{i=1}^{n}\big)$.

$\mathsf{Sign}(\mathsf{sk}, \boldsymbol{\tau}, (\boldsymbol{M_1}, \ldots, \boldsymbol{M_n}))$: to sign a vector $(M_1, \ldots, M_n) \in \mathbb{G}^n$ using $\mathsf{sk} = \big(h_z^{\alpha_r},\ \{\chi_i, \gamma_i, \delta_i\}_{i=1}^{n}\big)$ with the file identifier τ, conduct the following steps.

1. Choose $\theta \xleftarrow{R} \mathbb{Z}_p$ and compute

$$z = g_r^{\theta} \cdot \prod_{i=1}^{n} M_i^{-\chi_i} \qquad\qquad r = g_z^{-\theta} \cdot \prod_{i=1}^{n} M_i^{-\gamma_i} \qquad\qquad u = h_z^{-\theta \cdot \alpha_r} \cdot \prod_{i=1}^{n} M_i^{-\delta_i}$$

2. Using the bits $\tau[1] \ldots \tau[L]$ of $\tau \in \{0,1\}^L$, define the vector $\vec{f}_{\tau} = \vec{f}_{3,0} \cdot \prod_{i=1}^{L} \vec{f}_{3,i}^{\,\tau[i]}$ so as to assemble a Groth-Sahai CRS $\mathbf{f}_{\tau} = (\vec{f_1}, \vec{f_2}, \vec{f}_{\tau})$.
3. Using $\mathbf{f}_{\tau}$, compute Groth-Sahai commitments

$$\begin{aligned} \vec{C}_z &= (1_{\mathbb{G}}, 1_{\mathbb{G}}, z) \cdot \vec{f_1}^{\nu_{z,1}} \cdot \vec{f_2}^{\nu_{z,2}} \cdot \vec{f}_{\tau}^{\nu_{z,3}}, \\ \vec{C}_r &= (1_{\mathbb{G}}, 1_{\mathbb{G}}, r) \cdot \vec{f_1}^{\nu_{r,1}} \cdot \vec{f_2}^{\nu_{r,2}} \cdot \vec{f}_{\tau}^{\nu_{r,3}} \\ \vec{C}_u &= (1_{\mathbb{G}}, 1_{\mathbb{G}}, u) \cdot \vec{f_1}^{\nu_{u,1}} \cdot \vec{f_2}^{\nu_{u,2}} \cdot \vec{f}_{\tau}^{\nu_{u,3}} \end{aligned}$$

to z, r and u, respectively. Using the randomness of these commitments, generate proofs $\vec{\pi}_1 = (\pi_{1,1}, \pi_{1,2}, \pi_{1,3}) \in \mathbb{G}^3$ and $\vec{\pi}_2 = (\pi_{2,1}, \pi_{2,2}, \pi_{2,3}) \in \mathbb{G}^3$ that (z, r, u) satisfy the verification equations $1_{\mathbb{G}_T} = e(g_z, z) \cdot e(g_r, r) \cdot \prod_{i=1}^{n} e(g_i, M_i)$ and $1_{\mathbb{G}_T} = e(h_z, z) \cdot e(h, u) \cdot \prod_{i=1}^{n} e(h_i, M_i)$. These proofs are obtained as

$$\begin{aligned} \vec{\pi}_1 &= (\pi_{1,1}, \pi_{1,2}, \pi_{1,3}) = \big(g_z^{-\nu_{z,1}} \cdot g_r^{-\nu_{r,1}},\ g_z^{-\nu_{z,2}} \cdot g_r^{-\nu_{r,2}},\ g_z^{-\nu_{z,3}} \cdot g_r^{-\nu_{r,3}}\big) \\ \vec{\pi}_2 &= (\pi_{2,1}, \pi_{2,2}, \pi_{2,3}) = \big(h_z^{-\nu_{z,1}} \cdot h^{-\nu_{u,1}},\ h_z^{-\nu_{z,2}} \cdot h^{-\nu_{u,2}},\ h_z^{-\nu_{z,3}} \cdot h^{-\nu_{u,3}}\big) \end{aligned}$$

and satisfy the verification equations

$$\begin{aligned}\prod_{i=1}^{n} E\big(g_i, (1_{\mathbb{G}}, 1_{\mathbb{G}}, M_i)\big)^{-1} &= E\big(g_z, \vec{C}_z\big) \cdot E\big(g_r, \vec{C}_r\big) \cdot E(\pi_{1,1}, \vec{f}_1) \cdot E(\pi_{1,2}, \vec{f}_2) \cdot E(\pi_{1,3}, \vec{f}_\tau) \quad (9)\\ \prod_{i=1}^{n} E\big(h_i, (1_{\mathbb{G}}, 1_{\mathbb{G}}, M_i)\big)^{-1} &= E\big(h_z, \vec{C}_z\big) \cdot E\big(h, \vec{C}_u\big) \cdot E(\pi_{2,1}, \vec{f}_1) \cdot E(\pi_{2,2}, \vec{f}_2) \cdot E(\pi_{2,3}, \vec{f}_\tau).\end{aligned}$$

The signature consists of

$$\sigma = (\vec{C}_z, \vec{C}_r, \vec{C}_u, \vec{\pi}_1, \vec{\pi}_2) \in \mathbb{G}^{15}. \qquad (10)$$

$\mathsf{SignDerive}(\mathsf{pk}, \boldsymbol{\tau}, \{(\boldsymbol{\omega_i}, \boldsymbol{\sigma^{(i)}})\}_{\boldsymbol{i=1}}^{\boldsymbol{\ell}})$**:** given pk, a file identifier τ and ℓ tuples $(\omega_i, \sigma^{(i)})$, parse each signature $\sigma^{(i)}$ as a tuple of the form $\sigma^{(i)} = (\vec{C}_{z,i}, \vec{C}_{r,i}, \vec{C}_{u,i}, \vec{\pi}_{1,i}, \vec{\pi}_{2,i}) \in \mathbb{G}^{15}$ for $i = 1$ to ℓ. Otherwise, the derivation process proceeds in two steps.

1. Compute

$$\vec{C}_z = \prod_{i=1}^{\ell} \vec{C}_{z,i}^{\,\omega_i} \qquad \vec{C}_r = \prod_{i=1}^{\ell} \vec{C}_{r,i}^{\,\omega_i} \qquad \vec{C}_u = \prod_{i=1}^{\ell} \vec{C}_{u,i}^{\,\omega_i} \qquad \vec{\pi}_1 = \prod_{i=1}^{\ell} \vec{\pi}_{1,i}^{\,\omega_i} \qquad \vec{\pi}_2 = \prod_{i=1}^{\ell} \vec{\pi}_{2,i}^{\,\omega_i}$$

2. Re-randomize the above commitments and proofs using their homomorphic property and return the re-randomized version $\sigma = (\vec{C}_z, \vec{C}_r, \vec{C}_u, \vec{\pi}_1, \vec{\pi}_2)$.

$\mathsf{Verify}(\mathsf{pk}, \boldsymbol{\sigma}, \boldsymbol{\tau}, (\boldsymbol{M_1}, \ldots, \boldsymbol{M_n}))$**:** given a pair $(\tau, (M_1, \ldots, M_n))$ and a purported signature σ parse the latter as $(\vec{C}_z, \vec{C}_r, \vec{C}_u, \vec{\pi}_1, \vec{\pi}_2)$. Then, return 1 if and only if $(M_1, \ldots, M_n) \neq (1_{\mathbb{G}}, \ldots, 1_{\mathbb{G}})$ and equations (9) are satisfied.

We believe this construction to be of interest even if we disregard its structure-preserving property. Indeed, if we compare it with the only known completely context-hiding linearly homomorphic signature in the standard model [13], its signatures are shorter by one group element. Moreover, we can prove the security under the sole DLIN assumption whereas the scheme of [13] requires an additional assumption.

The scheme is clearly completely context hiding because signatures only consist of perfectly randomizable commitments and NIWI arguments.

As for the unforgeability of the scheme, the proof of the following theorem is along the lines of [55, Theorem 5]. However, we can only prove unforgeability in a weaker sense as we need to assume that the adversary is targeting. Namely, in the case of Type II attacks, the adversary must also output a proof that it actually broke the security of the scheme and that its vector $\vec{M}^\star = (M_1^\star, \ldots, M_n^\star) \in \mathbb{G}^n$ is indeed independent of the vectors for which it obtained signatures for the target tag $\tau^\star$.

If $\{\vec{M}_i = (M_{i,1}, \ldots, M_{i,n})\}_{i=1}^m$ denote the linearly independent vectors that were signed for $\tau^\star$, the adversary could simply output a vector $\vec{W} = (W_1, \ldots, W_n) \in \mathbb{G}^n$ such that $\prod_{j=1}^n e(M_j^\star, W_j) \neq 1_{\mathbb{G}_T}$ and $\prod_{j=1}^n e(M_{i,j}, W_j) = 1_{\mathbb{G}_T}$ for each $i \in \{1, \ldots, m\}$. The latter test guarantees that the adversary's output is a non-trivial Type II forgery.

Theorem 4. *The above scheme provides unforgeability against independent targeting adversaries if the DLIN assumption holds in* $\mathbb{G}$.

Proof. Since the scheme is completely context-hiding, we work with a simpler security definition where the adversary only interacts with a signing oracle. This suffices to guarantee security in the sense of Definition 2, as implied by the result of Ahn *et al.* [8]. The proof proceeds via a sequence of games. In each game, we denote by X_i the probability that the adversary $\mathcal{A}$ wins.

Game_{real} : This is the real game. When the adversary $\mathcal{A}$ terminates, the simulator outputs 1 if $\mathcal{A}$ is successful. We thus have $\Pr[X_{real}] = \mathbf{Adv}(\mathcal{A})$.

Game_0 : This game is identical to Game_{real} but we modify the generation of the public key. Namely, the vectors $(\vec{f_1}, \vec{f_2}, \{\vec{f_{3,i}}\}_{i=0}^{L})$ are chosen by setting $\vec{f_1} = (f_1, 1_{\mathbb{G}}, g)$ and $\vec{f_2} = (1_{\mathbb{G}}, f_2, g)$, with $f_1, f_2 \stackrel{R}{\leftarrow} \mathbb{G}$. As for $\{\vec{f_{3,i}}\}_{i=0}^{L}$, they are obtained as

$$\begin{aligned} \vec{f_{3,0}} &= \vec{f_1}^{\xi_{0,1}} \cdot \vec{f_2}^{\xi_{0,2}} \cdot (1,1,g)^{\xi_{0,3}} \cdot (1,1,g)^{\mu\cdot\zeta-\rho_0} \\ \vec{f_{3,i}} &= \vec{f_1}^{\xi_{i,1}} \cdot \vec{f_2}^{\xi_{i,2}} \cdot (1,1,g)^{\xi_{i,3}} \cdot (1,1,g)^{-\rho_i}, \qquad i \in \{1,\ldots,L\} \end{aligned} \tag{11}$$

with $\mu \stackrel{R}{\leftarrow} \{0,\ldots,L\}$, $\xi_{0,1}, \xi_{1,1}, \ldots, \xi_{L,1} \stackrel{R}{\leftarrow} \mathbb{Z}_p$, $\xi_{0,2}, \xi_{1,2}, \ldots, \xi_{L,2} \stackrel{R}{\leftarrow} \mathbb{Z}_p$, $\xi_{0,3}, \xi_{1,3}, \ldots, \xi_{L,3} \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and $\rho_0, \rho_1, \ldots, \rho_L \stackrel{R}{\leftarrow} \{0,\ldots,\zeta-1\}$, with $\zeta = 2q$ and where q is the number of distinct tags across all signing queries. Note that this change is only conceptual since $\{\vec{f_{3,i}}\}_{i=0}^{L}$ have the same distribution as in Game_{real}. We thus have $\Pr[X_0] = \mathbf{Adv}(\mathcal{A})$.

Game_1 : In this game, we first raise an event F_1, which causes the simulator $\mathcal{B}$ to abort if it does *not* occur. Let $\tau_1, \ldots, \tau_q$ be the distinct tags successively involved in $\mathcal{A}$'s queries throughout the game and let $\tau^\star$ be the tag involved in $\mathcal{A}$'s forgery. We know that, for a Type II forger, $\tau^\star \in \{\tau_1, \ldots, \tau_q\}$ whereas $\tau^\star \notin \{\tau_1, \ldots, \tau_q\}$ for a Type I adversary. For each string $\tau \in \{0,1\}^L$, we consider the function $J(\tau) = \mu \cdot \zeta - \rho_0 - \sum_{i=1}^{L} \rho_i \tau[i]$. We also define F_1 to be the event that

$$J(\tau^\star) = 0 \qquad \wedge \qquad \bigwedge_{\tau_j \in \{\tau_1,\ldots,\tau_q\}\setminus\{\tau^\star\}} J(\tau_j) \neq 0.$$

We note that the exponents $\rho_0, \rho_1, \ldots, \rho_L$ are independent of $\mathcal{A}$'s view: as a consequence, the simulator could equivalently define $\{\vec{f_{3,i}}\}_{i=0}^{L}$ first and only choose $\{\rho_i\}_{i=0}^{L}$ – together with values $\{\xi_{3,i}\}_{i=0}^{L}$ explaining the $\{\vec{f_{3,i}}\}_{i=0}^{L}$ – at the end of the game, when $\tau^\star, \tau_1, \ldots, \tau_q$ have been defined. In the case of a Type I attack, the same analysis as [60] (after the simplification of Bellare and Ristenpart [14]) shows that $\Pr[X_1 \wedge F_1] \geq \mathbf{Adv}(\mathcal{A})^2/(27 \cdot q \cdot (L+1))$.

This follows from the fact that, for any set of queries, a lower bound on the probability of event F_1 is $1/(2q(L+1))$. In the case of Type II attacks, a lower bound on the probability of F_1 for any set of queries is given by $\eta \geq 1/(2(q-1)(L+1)) > 1/(2q(L+1))$. Indeed, after re-ordering, the set of queried tags can be written $\{\tau^\star, \tau_1, \ldots, \tau_{q-1}\}$ and, from the known results [60, 50] on the programmability of Waters' hash function, we know that the probability, taken over the choice of $(\mu, \rho_0, \ldots, \rho_L)$, to have $J(\tau^\star) = 0$ and $\wedge_{j=1}^{q-1} J(\tau_j) \neq 0$ for any distinct $\tau^\star, \tau_1, \ldots, \tau_q$ is at least $1/(2(q-1)(L+1)) > 1/(2q(L+1))$. In the following, we denote by F_i the counterpart of event F_1 in Game_i.

Game_2 : In this game, we modify the distribution of the public key. Namely, $\vec{f_1} = (f_1, 1, g)$ and $\vec{f_2} = (1, f_2, g)$ are chosen as before but, instead of generating the vectors $\{\vec{f_{3,i}}\}_{i=0}^{L}$ as previously, we choose them as

$$\begin{aligned} \vec{f_{3,0}} &= \vec{f_1}^{\xi_{0,1}} \cdot \vec{f_2}^{\xi_{0,2}} \cdot (1,1,g)^{\mu\cdot\zeta-\rho_0} \\ \vec{f_{3,i}} &= \vec{f_1}^{\xi_{i,1}} \cdot \vec{f_2}^{\xi_{i,2}} \cdot (1,1,g)^{-\rho_i}, \qquad i \in \{1,\ldots,L\} \end{aligned} \tag{12}$$

which amounts to setting $\xi_{0,3} = \xi_{1,3} = \ldots = \xi_{L,3} = 0$. This change should not significantly affect $\mathcal{A}$'s behavior if the DLIN assumption holds. More precisely, if events $X_1 \wedge F_1$ and $X_2 \wedge F_2$ occur with noticeably different probabilities in Game_1 and Game_2, this contradicts the DLIN assumption. Concretely, consider a DLIN instance $(g, f_1, f_2, f_1^{\delta_1}, f_2^{\delta_2}, Z)$, where $\delta_1, \delta_2 \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and $Z = g^{\delta_1+\delta_2}$ or $Z \in_R \mathbb{G}$. Using the random self-reducibility of DLIN, we can create $L+1$ independent DLIN instances by picking $\varphi_i, \phi_i, \psi_i \stackrel{R}{\leftarrow} \mathbb{Z}_p$, for $i \in \{0, \ldots, L\}$ and setting

$$\vec{f}_{3,0} = \big((f_1^{\delta_1})^{\varphi_0} \cdot f_1^{\phi_0},\ (f_2^{\delta_2})^{\varphi_0} \cdot f_2^{\psi_0},\ Z^{\varphi_0} \cdot g^{\phi_0+\psi_0} \cdot (1,1,g)^{\mu\cdot\zeta-\rho_0}\big)$$
$$\vec{f}_{3,i} = \big((f_1^{\delta_1})^{\varphi_i} \cdot f_1^{\phi_i},\ (f_2^{\delta_2})^{\varphi_i} \cdot f_2^{\psi_i},\ Z^{\varphi_i} \cdot g^{\phi_i+\psi_i} \cdot (1,1,g)^{-\rho_i}\big), \qquad i \in \{1, \ldots, L\}$$

If $Z \in_R \mathbb{G}$, $\{\vec{f}_{3,i}\}_{i=0}^{L}$ is distributed as in Game_1. If $Z = g^{\delta_1+\delta_2}$, the distribution of $\{\vec{f}_{3,i}\}_{i=0}^{L}$ is the same as in (12). For this reason, we can write $|\Pr[X_2 \wedge F_2] - \Pr[X_1 \wedge F_1]| \leq \mathbf{Adv}^{\mathrm{DLIN}}(\mathcal{A})$ as we assumed that the challenger $\mathcal{B}$ can always detect when a targeting adversary is successful.

Game_3 : In this game, we modify the treatment of signing queries. We note that, for a given message $(\tau, \vec{M} = (M_1, \ldots, M_n))$, there is an exponential number of witnesses $(z, r, u) \in \mathbb{G}^3$ satisfying the verification equations

$$e(g_z, z) \cdot e(g_r, r) \cdot \prod_{i=1}^{n} e(g_i, M_i) = 1_{\mathbb{G}_T} \tag{13}$$
$$e(h_z, z) \cdot e(h, u) \cdot \prod_{i=1}^{n} e(h_i, M_i) = 1_{\mathbb{G}_T}.$$

Specifically, each $z \in_R \mathbb{G}$ determines a unique pair (r, u) for which (13) holds. However, in Game_3, the simulator $\mathcal{B}$ answers all signing queries using the witness (z, r, u) such that

$$z = \prod_{i=1}^{n} M_i^{-\chi_i} \qquad\qquad r = \prod_{i=1}^{n} M_i^{-\gamma_i} \qquad\qquad u = \prod_{i=1}^{n} M_i^{-\delta_i}$$

Note that this amounts to choosing $\theta = 0$ at step 1 of the signing algorithm. Still, $\mathcal{B}$ has a valid witness for the statement to be proved. It thus assembles a Groth-Sahai CRS $\mathbf{f} = (\vec{f}_1, \vec{f}_2, \vec{f}_\tau)$ by computing $\vec{f}_\tau = \vec{f}_{3,0} \cdot \prod_{i=1}^{L} \vec{f}_{3,i}^{\,\tau[i]}$. Using $\mathbf{f}$, it computes Groth-Sahai commitments $\vec{C}_z, \vec{C}_r, \vec{C}_u$ to z, r and u. Using the randomness of these commitments, it faithfully generates proofs $\vec{\pi}_1$ and $\vec{\pi}_2$ satisfying the verification equations (9).

We argue that this change does not affect $\mathcal{A}$'s view whatsoever. Indeed, if event F_3 occurs we have $J(\tau^\star) = 0$ and $J(\tau_j) \neq 0$ for each $\tau_j \neq \tau^\star$. Moreover, when $J(\tau_j)$, the Groth-Sahai CRS $(\vec{f}_1, \vec{f}_2, \vec{f}_{\tau_j})$ is a perfectly hiding Groth-Sahai CRS. This means that $\vec{C}_z$, $\vec{C}_r$, $\vec{C}_u$ are perfectly hiding commitments and proofs $(\vec{\pi}_1, \vec{\pi}_2)$ are perfectly witness indistinguishable proofs. In other words, although the proofs $(\vec{\pi}_1, \vec{\pi}_2)$ are always generated using the witnesses (z, r, u) for which $\theta = 0$, their distribution does not depend on which specific witness is used.

In contrast, in the case of Type II attacks, signing queries involving $\tau^\star$, $(\vec{C}_z, \vec{C}_r, \vec{C}_u, \vec{\pi}_1, \vec{\pi}_2)$ reveal the underlying (z, r, u) in the information theoretic sense since $(\vec{f}_1, \vec{f}_2, \vec{f}_{\tau^\star})$ is a perfectly binding CRS when $J(\tau^\star) = 0$. However, at most $n-1$ signing queries on linearly independent vectors $\vec{M}_j$ are made for the tag $\tau^\star$, so that $\mathcal{A}$ only obtains $n-1$ linearly independent equations in the exponent. As a consequence, $\mathcal{A}$ does not obtain a sufficient amount of information to recognize that $\theta = 0$ in the underlying signatures. For this reason, we find that $\Pr[X_3 \wedge F_3] = \Pr[X_2 \wedge F_2]$.

In Game_3, we show that a successful forger $\mathcal{A}$ implies an algorithm $\mathcal{B}$ solving a given SDP instance (g_z, g_r, h_z, h) with non-negligible advantage, which contradicts the DLIN assumption.

Recall that, when the adversary $\mathcal{A}$ terminates, it outputs $(\tau^\star, \vec{M}^\star, \sigma^\star)$, where $\vec{M}^\star = (M_1^\star, \ldots, M_n^\star)$ and $\sigma^\star = (\vec{C}_z^\star, \vec{C}_r^\star, \vec{C}_u^\star, \vec{\pi}_1^\star, \vec{\pi}_2^\star) \in \mathbb{G}^{15}$ satisfies the verification equations. At this point, if the event F_1 introduced in Game_1 occurs, we must have $J(\tau^\star) = 0$, which means that $\vec{f}_{\tau^\star} = \vec{f}_{3,0} \cdot \prod_{i=1}^{L+1} \vec{f}_{3,i}^{\,\tau^\star[i]}$ is in $\mathrm{span}(\vec{f}_1, \vec{f}_2)$. This implies that $\vec{C}_z^\star$, $\vec{C}_r^\star$ and $\vec{C}_u^\star$ are perfectly binding commitments. Moreover, using $(\log_g(f_1), \log_g(f_2))$, $\mathcal{B}$ can extract the underlying group elements $(z^\star, r^\star, u^\star) \in \mathbb{G}^3$ by performing

BBS decryptions of ciphertexts $(\vec{C}_z^\star, \vec{C}_r^\star, \vec{C}_u^\star)$. Since $(\vec{\pi}_1^\star, \vec{\pi}_2^\star)$ are valid proofs for a perfectly sound Groth-Sahai CRS, the extracted elements $(z^\star, r^\star, u^\star)$ necessarily satisfy

$$1_{\mathbb{G}_T} = e(g_z, z^\star) \cdot e(g_r, r^\star) \cdot \prod_{i=1}^{n} e(g_i, M_i^\star) = e(h_z, z^\star) \cdot e(h, u^\star) \cdot \prod_{i=1}^{n} e(h_i, M_i^\star). \tag{14}$$

Having extracted $(z^\star, r^\star, u^\star)$, $\mathcal{B}$ also computes

$$z^\dagger = \prod_{i=1}^{n} M_i^{\star\, -\chi_i} \qquad r^\dagger = \prod_{i=1}^{n} M_i^{\star\, -\gamma_i} \qquad u^\dagger = \prod_{i=1}^{n} M_i^{\star\, -\delta_i}, \tag{15}$$

so that $(z^\dagger, r^\dagger, u^\dagger)$ also satisfies (14). Since $(z^\dagger, r^\dagger, u^\dagger)$ and $(z^\star, r^\star, u^\star)$ both satisfy (14), the triple

$$(z^\ddagger, r^\ddagger, u^\ddagger) = \Big(\frac{z^\star}{z^\dagger}, \frac{r^\star}{r^\dagger}, \frac{u^\star}{u^\dagger}\Big)$$

necessarily satisfies $e(g_z, z^\ddagger) \cdot e(g_r, r^\ddagger) = e(h_z, z^\ddagger) \cdot e(h, u^\ddagger) = 1_{\mathbb{G}_T}$. To conclude the proof, we argue that $z^\ddagger \neq 1_{\mathbb{G}}$ with all but negligible probability.

To do this, we remark that, if the event F_1 defined in Game_1 occurs, the only information that $\mathcal{B}$ leaks about $(\chi_1, \ldots, \chi_n)$ resides in the unique signing query involving $\tau^\star$ if the case of Type II attacks. Indeed, for all signing queries $(\tau, \vec{M})$ involving tags τ such that $\tau \neq \tau^\star$, we have $J(\tau) \neq 0$ so that $(\vec{f_1}, \vec{f_2}, \vec{f_\tau})$ is a perfectly hiding Groth-Sahai CRS, for which proofs $(\vec{\pi}_1, \vec{\pi}_2)$ and commitments are perfectly witnesses indistinguishable. In other words, the signatures $(\vec{C}_z, \vec{C}_r, \vec{C}_u, \vec{\pi}_1, \vec{\pi}_2)$ for which $J(\tau) \neq 0$ leak nothing about $(\chi_1, \ldots, \chi_n)$. In contrast, in the case of Type II attacks, signing queries involving $\tau^\star$, $(\vec{C}_z, \vec{C}_r, \vec{C}_u, \vec{\pi}_1, \vec{\pi}_2)$ reveal the underlying (z, r, u) in the information theoretic sense. However, at most $n-1$ linearly independent vectors $\vec{M}_j$ are signed w.r.t. $\tau^\star$, so that $\mathcal{A}$ only obtains $n-1$ linearly independent equations in the exponent for the unknowns $(\chi_1, \ldots, \chi_n)$. As a consequence, we can apply the same arguments as in the proof of Theorem 1 and Lemma 1. With probability $1-1/p$, we thus have $z^\ddagger \neq z^\star$.

To recap, we find

$$\Pr[X_3 \wedge F_3] = \mathbf{Adv}^{\mathrm{SDP}}(\mathcal{B}) \cdot \Big(1 - \frac{1}{p}\Big)^{-1}.$$

When putting the above altogether, we find

$$\frac{\mathbf{Adv}(\mathcal{A})^2}{27 \cdot q \cdot (L+1)} \leq \mathbf{Adv}^{\mathrm{SDP}}(\mathcal{B}) \cdot \Big(1 - \frac{1}{p}\Big)^{-1} + \mathbf{Adv}^{\mathrm{DLIN}}(\mathcal{B}).$$

Since any SDP algorithm $\mathcal{B}_0$ yields a DLIN distinguisher $\mathcal{B}_1$ such that $\mathbf{Adv}^{\mathrm{DLIN}}(\mathcal{B}_0) \geq 2 \cdot \mathbf{Adv}^{\mathrm{SDP}}(\mathcal{B}_1)$, we find

$$\mathbf{Adv}(\mathcal{A}) \leq \sqrt{27 \cdot q \cdot (L+1) \cdot \Big[1 + \frac{1}{2} \cdot \Big(1 - \frac{1}{p}\Big)^{-1}\Big] \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\mathcal{B})}$$

and the announced result follows □

C Definitions for Trapdoor Commitments

Formally, a non-interactive commitment scheme $(\mathsf{Setup}, \mathsf{Com}, \mathsf{Verify})$ is a triple of probabilistic polynomial-time (PPT) algorithms where, on input of a security parameter λ, Setup outputs a public key pk; Com takes as input a message Msg, a public key pk and outputs a commitment/de-commitment pair $(\mathsf{com}, \mathsf{dec}) \overset{R}{\leftarrow} \mathsf{Com}(pk, \mathsf{Msg})$, and $\mathsf{Verify}(pk, \mathsf{Msg}, \mathsf{com}, \mathsf{dec})$ is deterministic and outputs 0 or 1.

The correctness property guarantees that Verify always outputs 1 whenever $(\mathsf{com}, \mathsf{dec})$ is obtained by committing to Msg using honestly generated parameters.

The *binding* property demands that, given pk, no PPT adversary should be able to produce a commitment that can be opened to two distinct messages. More precisely, for any PPT adversary $\mathcal{A}$, the following advantage function should be negligible as a function of λ.

$$\begin{aligned}\mathbf{Adv}^{\mathrm{bind}}_{\mathsf{CMT}}(\mathcal{A}) := \Pr[\ &\mathsf{Verify}(pk, \mathsf{Msg}_0, \mathsf{com}, \mathsf{dec}_0) = \mathsf{Verify}(pk, \mathsf{Msg}_1, \mathsf{com}, \mathsf{dec}_1) = 1 \ \wedge \\ &\mathsf{Msg}_0 \neq \mathsf{Msg}_1 \ : \ pk \xleftarrow{R} \mathsf{Setup}(\lambda);\ (\mathsf{com}, \mathsf{Msg}_0, \mathsf{dec}_0, \mathsf{Msg}_1, \mathsf{dec}_1) \xleftarrow{R} \mathcal{A}(pk)\ ]\end{aligned}$$

A commitment is also said *hiding* if commitment to distinct messages have computationally indistinguishable distributions. Formally, for any PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, the following advantage term is negligible as a function of λ.

$$\begin{aligned}\mathbf{Adv}^{\mathrm{hide}}_{\mathsf{CMT}}(\mathcal{A}) := \Big| \Pr[\ &b = b' \ : \ pk \xleftarrow{R} \mathsf{Setup}(\lambda);\ b \xleftarrow{R} \{0,1\};\ (\mathsf{Msg}_0, \mathsf{Msg}_1, st) \xleftarrow{R} \mathcal{A}_1(pk); \\ &(\mathsf{com}, \mathsf{dec}) \xleftarrow{R} \mathsf{Com}(pk, m_b);\ b' \xleftarrow{R} \mathcal{A}_2(\mathsf{com}, st)\ ] - \frac{1}{2} \Big|\end{aligned}$$

A trapdoor commitment is a perfectly hiding commitment for which a trapdoor tk makes it possible to break the binding property and open a commitment to any arbitrary value. However, this should remain infeasible without the trapdoor. More formally, a trapdoor commitment uses two additional algorithms $(\mathsf{FakeCom}, \mathsf{FakeOpen})$ that proceed as follows.

Definition 7. *A trapdoor commitment is a tuple* $(\mathsf{Setup}, \mathsf{Com}, \mathsf{FakeCom}, \mathsf{FakeOpen}, \mathsf{Verify})$ *of efficient algorithms where* Com *and* Verify *proceed as in an ordinary commitment and other algorithms proceed as follows.*

Setup: *is a randomized algorithm that takes as input a security parameter λ. It produces a public key pk and a trapdoor tk.*

FakeCom: *is a randomized algorithm that takes as input a public key pk and the trapdoor tk. It outputs a fake commitment string* $\widetilde{\mathsf{com}}$ *and some auxiliary information* aux.

FakeOpen: *takes as input a fake commitment produced by* $\mathsf{FakeCom}$ *and the corresponding auxiliary information* aux. *It also takes as input a message* Msg *and the trapdoor tk and outputs a fake de-commitment* $\widetilde{\mathsf{dec}}$ *such that* $\mathsf{Verify}(pk, \mathsf{Msg}, \widetilde{\mathsf{com}}, \widetilde{\mathsf{dec}}) = 1$. *Moreover, the two distributions*

$$\begin{aligned}D_{fake} := \{(pk, tk) \leftarrow \mathsf{Setup}(\lambda);\ &(\widetilde{\mathsf{com}}, \mathsf{aux}) \leftarrow \mathsf{FakeCom}(pk, tk); \\ &\widetilde{\mathsf{dec}} \leftarrow \mathsf{FakeOpen}(\mathsf{aux}, tk, \widetilde{\mathsf{com}}, \mathsf{Msg}) : (pk, \mathsf{Msg}, \widetilde{\mathsf{com}}, \widetilde{\mathsf{dec}})\}\end{aligned}$$

and

$$D_{real} := \{(pk, tk) \leftarrow \mathsf{Setup}(\lambda);\ (\mathsf{com}, \mathsf{dec}) \leftarrow \mathsf{Com}(pk, \mathsf{Msg}) : (pk, \mathsf{Msg}, \mathsf{com}, \mathsf{dec})\}$$

should be indistinguishable.

We now recall the definition of independence for commitment schemes, which is known (see, e.g., [44] for a proof) to imply re-usable non-malleability with respect to opening.

Definition 8 ([32]). *A trapdoor commitment scheme* $(\mathsf{Setup}, \mathsf{Com}, \mathsf{FakeCom}, \mathsf{FakeOpen}, \mathsf{Verify})$ *provides ℓ-independence if, for any PPT adversary $(\mathcal{A}_1, \mathcal{A}_2)$ and any pair of ℓ-tuples* $(\mathsf{Msg}_1, \ldots, \mathsf{Msg}_\ell)$,

$(\mathsf{Msg}'_1,\ldots,\mathsf{Msg}_\ell)'$, *the following probability is a negligible function of the security parameter* λ*:*

$$
\begin{aligned}
\Pr[\,&(pk,tk)\leftarrow \mathsf{Setup}(\lambda);\ R_1,\ldots,R_\ell \xleftarrow{R} \{0,1\}^{\mathsf{poly}(\lambda)};\\
&(\widetilde{\mathsf{com}}_i,\mathsf{aux}_i)\leftarrow \mathsf{FakeCom}(pk,tk,R_i)\\
&(st,\mathsf{com}^\star)\leftarrow \mathcal{A}_1(pk,\widetilde{\mathsf{com}}_1,\ldots,\widetilde{\mathsf{com}}_\ell)\ \textit{with}\ \mathsf{com}^\star\not\in\{\widetilde{\mathsf{com}}_i\}_{i=1}^{\ell}\\
&\mathsf{dec}_i\leftarrow \mathsf{FakeOpen}(\mathsf{aux}_i,tk,\widetilde{\mathsf{com}}_i,\mathsf{Msg}_i)\quad \forall i\in\{1,\ldots,\ell\}\\
&\mathsf{dec}'_i\leftarrow \mathsf{FakeOpen}(\mathsf{aux}_i,tk,\widetilde{\mathsf{com}}_i,\mathsf{Msg}'_i)\quad \forall i\in\{1,\ldots,\ell\}\\
&(\mathsf{Msg}^\star_1,\mathsf{dec}^\star_1)\leftarrow \mathcal{A}_2(st,pk,\mathsf{Msg}_1,\mathsf{dec}_1,\ldots,\mathsf{Msg}_\ell,\mathsf{dec}_\ell)\\
&(\mathsf{Msg}^\star_2,\mathsf{dec}^\star_2)\leftarrow \mathcal{A}_2(st,pk,\mathsf{Msg}'_1,\mathsf{dec}'_1,\ldots,\mathsf{Msg}'_\ell,\mathsf{dec}'_\ell):\\
&\mathsf{Msg}^\star_1\neq\mathsf{Msg}^\star_2\wedge \mathsf{Verify}(pk,\mathsf{Msg}^\star_1,\mathsf{com}^\star,\mathsf{dec}^\star_1)=1\wedge \mathsf{Verify}(pk,\mathsf{Msg}^\star_2,\mathsf{com}^\star,\mathsf{dec}^\star_2)=1]
\end{aligned}
$$

A trapdoor commitment is independent *if it provides* ℓ*-independence for any arbitrary* $\ell\in\mathsf{poly}(\lambda)$.

It is known (see, *e.g.*, [54]) that, when a SSTC scheme and a secure one-time signature are combined to build an ordinary commitment scheme, the simulation-sound binding property and the security of the one-time signature imply the notion of independence.

D Proof of Theorem 3

Proof. We first observe that the commitment satisfies the trapdoor property if the homomorphic SPS is regular. Indeed, in the distribution D_{fake}, the commitment $\widetilde{\mathsf{com}}$ is obtained as

$$c_j=\prod_{\mu=1}^{n_z} e(F_{j,\mu},\hat{Z}_\mu)\cdot\prod_{\nu=1}^{n_v} e(T_{j,\nu},\hat{V}_\nu)\cdot\prod_{i=1}^{n} e(G_{j,i},\hat{M}_i)\qquad j\in\{1,\ldots,m\}\tag{16}$$

where $(\hat{M}_1,\ldots,\hat{M}_n)\in_R\mathbb{G}^n$ and for a uniformly random tuple $(\hat{Z}_1,\ldots,\hat{Z}_{n_z},\hat{V}_1,\ldots,\hat{V}_{n_v})\in_R\mathbb{G}^{n_z+n_v}$. We also know that, for any $(M_1,\ldots,M_n)\neq(\hat{M}_1,\ldots,\hat{M}_n)$, the vector $(M_1/\hat{M}_1,\ldots,M_n/\hat{M}_n)$ has a valid signature $\sigma'=(Z'_1,\ldots,Z'_{n_z},V'_1,\ldots,V'_{n_v})$, so that there exists

$$\widetilde{\mathsf{dec}}=(\tilde{Z}_1,\ldots,\tilde{Z}_{n_z},\tilde{V}_1,\ldots,\tilde{V}_{n_v})=(Z'_1\cdot\hat{Z}_1,\ldots,\hat{Z}'_{n_z}\cdot\hat{Z}_{n_z},V'_1\cdot\hat{V}_1,\ldots,V'_{n_v}\cdot\hat{V}_{n_v})$$

that explains $\widetilde{\mathsf{com}}$ as a commitment to $(M_1,\ldots,M_n)$. Moreover, since $(\hat{Z}_1,\ldots,\hat{Z}_{n_v},\hat{V}_1,\ldots,\hat{V}_{n_v})$ was chosen uniformly in $\mathbb{G}^{n_z+n_v}$, $\widetilde{\mathsf{dec}}$ is uniform among values $(\tilde{Z}_1,\ldots,\tilde{Z}_{n_z},\tilde{V}_1,\ldots,\tilde{V}_{n_v})$ such that

$$c_j=\prod_{\mu=1}^{n_z} e(F_{j,\mu},\tilde{Z}_\mu)\cdot\prod_{\nu=1}^{n_v} e(T_{j,\nu},\tilde{V}_\nu)\cdot\prod_{i=1}^{n} e(G_{j,i},M_i)\qquad j\in\{1,\ldots,m\}.\tag{17}$$

In other words, the joint distribution of $(\widetilde{\mathsf{com}},\widetilde{\mathsf{dec}})$ is the same as if it were obtained by choosing $(\tilde{Z}_1,\ldots,\tilde{Z}_{n_z},\tilde{V}_1,\ldots,\tilde{V}_{n_v})\xleftarrow{R}\mathbb{G}^{n_v+n_z}$ and computing $\{c_j\}_{j=1}^m$ as per (17).

We now turn to the simulation-sound binding property and show that, if there exists a PPT adversary $\mathcal{A}$ that breaks this property with non-negligible advantage ε, there exits a non-independent Type I forger $\mathcal{B}$ against the signature scheme.

Concretely, our adversary $\mathcal{B}$ obtains a public key pk from its own challenger and sends the commitment key $pk=\mathsf{pk}$ to $\mathcal{A}$. Whenever $\mathcal{A}$ sends a query (commit,tag) to the $\mathcal{O}_{tk,pk}$ oracle, $\mathcal{B}$ faithfully runs the $\mathsf{SSTC.FakeCom}$ algorithm and thus computes $\widetilde{\mathsf{com}}=\{\tilde{c}_j\}_{j=1}^m$ according to (16) for randomly chosen $(\hat{M}_1,\ldots,\hat{M}_n)\xleftarrow{R}\mathbb{G}^n$, $\hat{\mathsf{dec}}=(\hat{Z}_1,\ldots,\hat{Z}_{n_z},\hat{V}_1,\ldots,\hat{V}_{n_v})\xleftarrow{R}\mathbb{G}^{n_z+n_v}$ and retains the information $\mathsf{aux}=((\hat{M}_1,\ldots,\hat{M}_n),\hat{\mathsf{dec}})$. When the oracle $\mathcal{O}_{tk,pk}$ subsequently receives a query of the form $(\mathsf{decommit},\widetilde{com},(M_1,\ldots,M_n))$, the reduction $\mathcal{B}$ invokes its own signing oracle on the input $(tag, (M_1/\hat{M}_1, \ldots, M_n/\hat{M}_n))$. Upon receiving the resulting signature $(Z'_1, \ldots, Z'_{n_z}, V'_1, \ldots, V'_{n_v})$, $\mathcal{B}$ computes and returns $\widetilde{\mathsf{dec}} = (\hat{Z}_1 \cdot Z'_1, \ldots, \hat{Z}_{n_z} \cdot Z'_{n_z}, \hat{V}_1 \cdot V'_1, \ldots, \hat{V}_{n_v} \cdot V'_{n_v})$.

Eventually, the adversary $\mathcal{A}$ outputs a commitment of its own $\mathsf{com}^\star = (c_1^\star, \ldots, c_m^\star)$ along with valid openings $\mathsf{dec} = (Z_1, \ldots, Z_{n_z}, V_1, \ldots, V_{n_v})$, $\mathsf{dec}' = (Z'_1, \ldots, Z'_{n_z}, V'_1, \ldots, V'_{n_v})$ to distinct vectors $(M_1, \ldots, M_n) \neq (M'_1, \ldots, M'_n)$ for some tag $tag^\star$ that has never been used in *any* query to $\mathcal{O}_{tk,sk}$. Since both openings successfully pass the verification test, we find that

$$\big(Z_1/Z'_1, \ldots, Z_{n_z}/Z'_{n_z}, \ldots, V_1/V'_1, \ldots, V_{n_v}/V'_{n_v}\big)$$

forms a valid homomorphic signature on the vector $(M_1/M'_1, \ldots, M_n/M'_n) \neq (1_{\mathbb{G}}, \ldots, 1_{\mathbb{G}})$ for the identifier $\tau^\star = tag^\star$. By construction, $\tau^\star$ was never the input of a signing query made by $\mathcal{B}$ to its own oracle. Consequently, $\mathcal{B}$ is indeed a Type I non-independent forger with advantage ε. □

E Non-Interactive Simulation-Sound Trapdoor Commitments from Linearly Homomorphic Signatures in Groups of Public Order

MacKenzie and Yang [54] showed that simulation-sound trapdoor commitments imply digital signatures. In the converse direction, constructions of SSTCs are only known for signature schemes admitting efficient Σ protocols. In fact, as noted by Fujisaki [39], all known constructions of non-interactive simulation-sound or multi-trapdoor [41] commitments build on signature schemes for which an efficient Σ protocol allows proving knowledge of a signature.

The idea is to commit to a message m by using m as the challenge of a Σ protocol for proving knowledge of a signature $\sigma = \mathsf{Sig}(sk, tag)$ on the tag. The commitment is given by the first message a of the Σ protocol transcript (a, m, z), which is obtained by simulating a proof of knowledge of a valid signature σ on the message tag. The commitment is subsequently opened by revealing z. By the special soundness of the Σ protocol, unless the sender actually knows a valid signature on tag, it can only open a given commitment a to one message m.

While simple, the above construction (which extends to give identity-based trapdoor commitments, as noted in [24]) does not readily extend to commit to vectors. Fujisaki [39] gave an alternative construction based on encryption schemes. However, this construction is interactive. Groth and Ostrovsky [46] finally defined the notion of simulation-extractable commitments by additionally requiring adversarially-generated commitments to be extractable instead of simply binding. A consequence of this strengthened property is that, just like UC commitments [25], simulation-extractable commitments cannot be length-reducing any longer.

This section shows that ordinary (*i.e.*, non-structure-preserving) linearly homomorphic signatures also make it possible to construct non-interactive simulation-sound (and thus non-malleable) commitments if they satisfy a certain template. Moreover, they make it possible to commit to vectors while preserving the ability of efficiently proving properties about committed vectors. We notably obtain efficient constructions based on the Diffie-Hellman and strong Diffie-Hellman [15] assumptions.

E.1 Definition and Template

We first consider a definition of unforgeability which is obtained by simplifying Definition 2 and removing the $\mathsf{SignDerive}$ and Reveal oracles. As we will see, this simplified definition will be sufficient for the construction of simulation-sound trapdoor commitments. On the other hand, unlike the definition used in [17–19], Definition 9 allows the adversary to choose the file identifiers in his signing queries.

Definition 9. *A linearly homomorphic signature scheme* $\Sigma = (\mathsf{Keygen}, \mathsf{Sign}, \mathsf{SignDerive}, \mathsf{Verify})$ *is secure if no probabilistic polynomial time (PPT) adversary has non-negligible advantage (as a function of the security parameter* $\lambda \in \mathbb{N}$*) in the following game:*

1. *The adversary $\mathcal{A}$ chooses an integer $n \in \mathbb{N}$ and sends it to the challenger who runs* $\mathsf{Keygen}(\lambda, n)$ *and obtains* $(\mathsf{pk}, \mathsf{sk})$ *before sending* pk *to $\mathcal{A}$.*
2. *On a polynomial number of occasions, $\mathcal{A}$ chooses a tag $\tau \in \mathcal{T}$ and a vector $\vec{v}$. The challenger returns $\sigma = \mathsf{Sign}(\mathsf{sk}, \tau, \vec{v})$ to $\mathcal{A}$.*
3. *$\mathcal{A}$ outputs an identifier $\tau^\star$, a signature $\sigma^\star$ and a vector $\vec{y} \in \mathbb{Z}_N^n$. The adversary $\mathcal{A}$ is deemed successful if* $\mathsf{Verify}(\mathsf{pk}, \tau^\star, \vec{y}^\star, \sigma^\star) = 1$ *and either of the following holds:*
   - ◦ *(Type I): $\tau^\star \neq \tau_i$ for any i and $\vec{y}^\star \neq \vec{0}$.*
   - ◦ *(Type II): $\tau^\star = \tau_i$ for some $i \in \{1, \ldots, q\}$ and $\vec{y}^\star \notin V_i$, where V_i denotes the subspace spanned by all vectors $\vec{v}_1, \ldots, \vec{v}_{k_i}$ that have been queried for τ_i.*

Note that, in some cases, it may be sufficient to use a *non-adaptive* definition of unforgeability where the adversary has to declare all the file identifier $\tau_1, \ldots, \tau_q$ involved in signing queries at the very beginning of the attack (before seeing the public key pk).

Again, we say that the adversary is *independent* if

- For any given tag τ, it is restricted to only query signatures on linearly independent vectors.
- Each pair $(\tau, \vec{m})$ is queried at most once.

Let $\Pi = (\mathsf{Keygen}, \mathsf{Sign}, \mathsf{SignDerive}, \mathsf{Verify})$ be a linearly homomorphic signature over $\mathbb{Z}_p^n$, for some large prime $p > 2^\lambda$. We assume that Π uses groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of public orders p^k and p, respectively, for some $k \in \mathbb{N}$. We also assume that each signature σ lives in $\mathbb{G}_1$. The verification algorithm takes as input a purported signature $\sigma \in \mathbb{G}_1$, a file identifier τ and a vector $\vec{m}$. It returns 1 if and only if

$$F(\sigma, \vec{m}, \mathsf{pk}, \tau) = 1_{\mathbb{G}_2}, \tag{18}$$

where F is a function ranging over the group $\mathbb{G}_2$ and satisfying certain linearity properties. Namely, for each pk produced by Keygen and each τ, we require that

$$F(\sigma_1 \cdot \sigma_2, \vec{m}_1 + \vec{m}_2, \mathsf{pk}, \tau) = F(\sigma_1, \vec{m}_1, \mathsf{pk}, \tau) \cdot F(\sigma_2, \vec{m}_2, \mathsf{pk}, \tau)$$

for any vectors $\vec{m}_1, \vec{m}_2 \in \mathbb{Z}_p^n$ and any $\sigma_1, \sigma_2 \in \mathbb{G}_1$. As a consequence, we also have

$$F(\sigma, \vec{m}, \mathsf{pk}, \tau)^\omega = F(\sigma^\omega, \omega \cdot \vec{m}, \mathsf{pk}, \tau)$$

for any $\omega \in \mathbb{Z}_p$ and any $\sigma \in \mathbb{G}_1$. Finally, the derivation algorithm $\mathsf{SignDerive}$ proceeds by computing $\mathsf{SignDerive}(\mathsf{pk}, \tau, \{(\omega_i, \sigma^{(i)})\}_{i=1}^{\ell}) = \prod_{i=1}^{\ell} \sigma^{(i)^{\omega_i}}$.

We remark that the above template only captures schemes in groups of public order, so that constructions based on the Strong RSA assumption [26, 27] or on lattices [18, 19] are not covered. The reason is that, when working over the integers, messages and signature components may increase at each homomorphic operation. This makes it harder to render trapdoor openings indistinguishable from original de-commitments.

E.2 Simulation-sound Trapdoor Commitments from Linearly Homomorphic Signatures

From a linearly homomorphic signature scheme $\Pi = (\mathsf{Keygen}, \mathsf{Sign}, \mathsf{SignDerive}, \mathsf{Verify})$ satisfying the template of Appendix E.1, we construct a non-interactive length-reducing SSTC as follows.

SSTC.Setup(λ, n): given the required dimension $n \in \mathbb{N}$ of committed vectors, run $\Pi.\mathsf{Keygen}(\lambda, n)$ to obtain a public key pk and a private key sk. The commitment key is $pk = \mathsf{pk}$ and the trapdoor tk consists of the private key sk of Π.

SSTC.Com($\boldsymbol{pk, tag, \vec{m}}$): to commit to a vector $\vec{m} \in \mathbb{Z}_p^n$, choose $\sigma \xleftarrow{R} \mathbb{G}_1$ in the signature space. Compute and output

$$c = F(\sigma, \vec{m}, \mathsf{pk}, tag)$$

by evaluating F as in the left-hand-side member of the verification equation (18). The commitment string is $\mathsf{com} = c$ whereas the decommitment is $\mathsf{dec} = \sigma$.

SSTC.FakeCom($\boldsymbol{pk, tk, tag}$): proceeds identically to SSTC.Com but using a randomly chosen vector $\vec{m}_{fake} \xleftarrow{R} \mathbb{Z}_p^n$. If $(\hat{\mathsf{com}}, \hat{\mathsf{dec}})$ denotes the resulting commitment/decommitment pair, the algorithms sets $\widetilde{\mathsf{com}} = \hat{\mathsf{com}}$ and $\mathsf{aux} = (\vec{m}_{fake}, \hat{\mathsf{dec}})$.

SSTC.FakeOpen(aux, $\boldsymbol{tk, tag}$, $\widetilde{\mathbf{com}}$, $\boldsymbol{\vec{m}}$): the algorithm parses $\widetilde{com}$ as $\tilde{c} \in \mathbb{G}_2$ and aux as $\big(\vec{m}_{fake}, \hat{\mathsf{dec}})$, where $\hat{\mathsf{dec}} = \hat{\sigma} \in \mathbb{G}_1$. It first generates a linearly homomorphic signature on the difference vector $\vec{m} - \vec{m}_{fake} \in \mathbb{Z}_p^n$ for the tag $tag = \tau$. Namely, using the trapdoor $tk = \mathsf{sk}$, compute

$$\sigma' \leftarrow \mathit{\Pi}.\mathsf{Sign}\big(\mathsf{sk}, \tau, \vec{m} - \vec{m}_{fake}\big).$$

Finally, it computes $\widetilde{\sigma} = \mathsf{SignDerive}(\mathsf{pk}, \tau, \{(1, \hat{\sigma}), (1, \sigma')\}) = \hat{\sigma} \cdot \sigma' \in \mathbb{G}_1$ and returns $\widetilde{\mathsf{dec}} = \widetilde{\sigma}$.

SSTC.Verify($\boldsymbol{pk, tag, \vec{m}}$, com, dec): parse the commitment com as $c \in \mathbb{G}_2$ and the opening dec as $\sigma \in \mathbb{G}_1$. If these cannot be parsed properly, return 0. Otherwise, return 1 if $c = F(\sigma, \vec{m}, \mathsf{pk}, tag)$ and 0 otherwise.

For completeness, we prove the following result in a similar way to the proof of Theorem 3.

Theorem 5. *The above construction is a secure SSTC assuming that Π is both regular and unforgeable against non-independent Type I attacks.*

Proof. The proof is very similar to the proof of Theorem 3. We first show that the commitment is a trapdoor commitment if Π is a regular homomorphic signature. Indeed, in the distribution D_{fake}, the commitment is obtained as

$$\widetilde{\mathsf{com}} = F(\hat{\sigma}, \vec{m}_{fake}, \mathsf{pk}, tag) \tag{19}$$

where $\vec{m}_{fake} \in_R \mathbb{Z}_p^n$ and $\hat{\sigma} \in_R \mathbb{G}_1$. Since Π is regular, we also know that, for any $\vec{m} \neq \vec{m}_{fake}$, the vector $\vec{m} - \vec{m}_{fake}$ has a valid signature $\sigma' \in \mathbb{G}_1$. As a consequence, there exists

$$\widetilde{\mathsf{dec}} = \tilde{\sigma} = \mathsf{SignDerive}(\mathsf{pk}, \tau, \{(1, \hat{\sigma}), (1, \sigma')\}) = \hat{\sigma} \cdot \sigma'$$

such that $\widetilde{\mathsf{com}} = F(\tilde{\sigma}, \vec{m}, \mathsf{pk}, tag)$, so that $\widetilde{\mathsf{com}}$ can be explained as a commitment to $\vec{m}$. Moreover, since $\hat{\sigma}$ was chosen uniformly in $\mathbb{G}_1$, the obtained de-commitment $\tilde{\sigma}$ is uniform among values such that

$$\widetilde{\mathsf{com}} = F(\tilde{\sigma}, \vec{m}, \mathsf{pk}, tag)$$

Said otherwise, $(\widetilde{\mathsf{com}}, \widetilde{\mathsf{dec}})$ has the same distribution as if it were obtained by choosing $\widetilde{dec} = \tilde{\sigma} \xleftarrow{R} \mathbb{G}_1$ and computing $\widetilde{\mathsf{com}} = F(\tilde{\sigma}, \vec{m}, \mathsf{pk}, tag)$.

To establish the simulation-sound binding property, we show that, if there exists a PPT adversary $\mathcal{A}$ that breaks this property with advantage ε, the homomorphic signature scheme Π can be broken by a non-independent Type I forger $\mathcal{B}$ with the same advantage ε.

Algorithm $\mathcal{B}$ takes as input a linearly homomorphic signature public key pk and sends $pk = \mathsf{pk}$ to the simulation-binding adversary $\mathcal{A}$. When $\mathcal{A}$ sends a query (commit, tag) to the $\mathcal{O}_{tk,pk}$ oracle, $\mathcal{B}$ runs the SSTC.FakeCom algorithm and computes $\widetilde{\mathsf{com}} = F(\hat{\sigma}, \vec{m}_{fake}, \mathsf{pk}, tag)$ for randomly chosen $\hat{\sigma} \xleftarrow{R} \mathbb{G}_1$ and $\vec{m}_{fake} \xleftarrow{R} \mathbb{Z}_p^n$. It retains the state information $\mathsf{aux} = (\vec{m}_{fake}, \hat{\sigma})$. For each invocation of the oracle $\mathcal{O}_{tk,pk}$ for an input of the form $(\mathsf{decommit}, \widetilde{com}, \vec{m})$, $\mathcal{B}$ sends the query $(tag, \vec{m} - \vec{m}_{fake})$ to its own signing oracle. Upon receiving the latter's response σ', $\mathcal{B}$ computes and returns $\widetilde{\mathsf{dec}} = \sigma' \cdot \hat{\sigma}$.

Eventually, $\mathcal{A}$ comes up with a commitment of its own $\mathsf{com}^\star$ with valid openings $\mathsf{dec} = \sigma$, $\mathsf{dec}' = \sigma'$ to distinct vectors $\vec{m} \neq \vec{m}'$ for a tag $tag^\star$ that it never submitted to $\mathcal{O}_{tk,sk}$. Since $\vec{m} \neq \vec{m}'$ and dec and dec' are valid openings of $\mathsf{com}^\star$ to $\vec{m}$ and $\vec{m}'$, respectively, the triple

$$\big(\tau^\star, \sigma/\sigma', \vec{m} - \vec{m}'\big)$$

forms a valid Type I forgery for the linearly homomorphic scheme $\mathit{\Pi}$. □

E.3 Instantiations

Construction from the Diffie-Hellman Assumption. Previously, non-malleable commitments based on the CDH assumption were — implicitly or explicitly — described in [34, 57] but it is not immediate how to extend them to commit to vectors in a modular way.

In [12], Attrapadung *et al.* described a linearly homomorphic signature which is notably secure against Type I independent adversaries — as implicitly proved by [12, Lemma 8] — under the computational Diffie-Hellman (CDH) assumption.

Keygen(λ, n): given a security parameter $\lambda \in \mathbb{N}$ and an integer $n \in \mathsf{poly}(\lambda)$, choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$. Choose $\alpha \xleftarrow{R} \mathbb{Z}_p$, $g, v \xleftarrow{R} \mathbb{G}$ and $u_0, u_1, \ldots, u_L \xleftarrow{R} \mathbb{G}$, for some $L \in \mathsf{poly}(\lambda)$. These elements $(u_0, \ldots, u_L) \in \mathbb{G}^{L+1}$ will be used to implement a programmable hash function $H_{\mathbb{G}} : \{0,1\}^L \to \mathbb{G}$ such that any L-bit string $\tau = \tau[1] \ldots \tau[L] \in \{0,1\}^L$ is mapped to the hash value $H_{\mathbb{G}}(\tau) = u_0 \cdot \prod_{i=1}^{L} u_i^{\tau[i]}$. Pick $g_i \xleftarrow{R} \mathbb{G}$ for $i = 1$ to n. Finally, define the identifier space $\mathcal{T} := \{0,1\}^L$. The private key is $\mathsf{sk} := \alpha$ and the public key consists of

$$\mathsf{pk} := \Big((\mathbb{G}, \mathbb{G}_T),\ g,\ g^\alpha,\ v,\ \{g_i\}_{i=1}^{n},\ \{u_i\}_{i=0}^{L}\Big).$$

Sign(sk, $\tau, \vec{m}$): given a vector $\vec{m} = (m_1, \ldots, m_n) \in \mathbb{Z}_p^n$, a file identifier $\tau \in \{0,1\}^L$ and the private key $\mathsf{sk} = \alpha \in \mathbb{Z}_p$, return $\perp$ if $\vec{m} = \vec{0}$. Otherwise, choose $r, s \xleftarrow{R} \mathbb{Z}_p$. Then, compute a signature $\sigma = (\sigma_1, \sigma_2, s) \in \mathbb{G}^2 \times \mathbb{Z}_p$ as

$$\sigma_1 = (g_1^{m_1} \cdots g_n^{m_n} \cdot v^s)^\alpha \cdot H_{\mathbb{G}}(\tau)^r, \qquad \sigma_2 = g^r.$$

SignDerive(pk, $\tau, \{(\beta_i, \sigma_i)\}_{i=1}^{\ell}$): given pk, a file identifier τ and ℓ tuples (β_i, σ_i), parse each signature σ_i as $\sigma_i = (\sigma_{i,1}, \sigma_{i,2}, s_i)$ for $i = 1$ to ℓ. Then, choose $\tilde{r} \xleftarrow{R} \mathbb{Z}_p$ and compute

$$\sigma_1 = \prod_{i=1}^{\ell} \sigma_{i,1}^{\beta_i} \cdot H_{\mathbb{G}}(\tau)^{\tilde{r}} \qquad \sigma_2 = \prod_{i=1}^{\ell} \sigma_{i,2}^{\beta_i} \cdot g^{\tilde{r}} \qquad s = \sum_{i=1}^{\ell} \beta_i \cdot s_i$$

and output (σ_1, σ_2, s).

Verify(pk, $\tau, \vec{m}, \sigma$): given pk, a signature $\sigma = (\sigma_1, \sigma_2, s)$ and a message $(\tau, \vec{m})$, where $\tau \in \{0,1\}^L$ and $\vec{m}$ is a vector $(m_1, \ldots, m_n) \in (\mathbb{Z}_p)^n$, return 0 if $\vec{m} = \vec{0}$. Otherwise, return 1 if

$$e(\sigma_1, g) = e(g_1^{m_1} \cdots g_n^{m_n} \cdot v^s, g^\alpha) \cdot e(H_{\mathbb{G}}(\tau), \sigma_2).$$

and 0 otherwise.

This scheme can be seen as a specific instantiation of the template where the group $\mathbb{G}_1$ is a product $\mathbb{G}_1 = \mathbb{G}^2 \times \mathbb{Z}_p$, which is a group for the operation $(\cdot, \cdot, +)$, and $\mathbb{G}_2 = \mathbb{G}_T$. Here, $\mathbb{G}_1$ and $\mathbb{G}_2$ thus have order p^3 and p, respectively. As for the linear function F, it can be instantiated as

$$F\big((\sigma_1, \sigma_2, s), \vec{m}, \mathsf{pk}, \tau\big) := e(\sigma_1, g^{-1}) \cdot e(H_{\mathbb{G}}(\tau), \sigma_2) \cdot e(g_1^{m_1} \cdots g_n^{m_n} \cdot v^s, g^\alpha)$$

As a result, we obtain a new non-interactive simulation-sound trapdoor commitment to vectors under the CDH assumption. We note that the scheme can be optimized by removing the terms v^s and s, so as to have $(\sigma_1, \sigma_2) = \big((\prod_{i=1}^n g_i^{m_i})^\alpha \cdot H_{\mathbb{G}}(\tau)^r, g^r \big)$ and

$$F\big((\sigma_1,\sigma_2), \vec{m}, \mathsf{pk}, \tau\big) := e(\sigma_1, g^{-1}) \cdot e(H_{\mathbb{G}}(\tau), \sigma_2) \cdot e(g_1^{m_1} \cdots g_n^{m_n}, g^\alpha)$$

Indeed, in the proof of Lemma 8 in [12], we observe that, if the signature scheme only needs to be secure against Type I attacks, the terms $(v^s, s) \in \mathbb{G} \times \mathbb{Z}_p$ can be eliminated.

Unlike the CDH-based construction of [39], the above commitment scheme is non-interactive and allows committing to vectors with a constant-size commitment string. Unlike the solution consisting in committing to a short string obtained by hashing the vector, our solution makes it possible for the sender to prove properties (using Σ protocols or Groth-Sahai proofs) about committed vectors in an efficient way.

We also remark that, for vectors of dimension $n = 1$, we obtain a simplification of existing multi-trapdoor (or identity-based) trapdoor commitments [34, 57] based on the Waters signature: instead of starting from a Σ protocol for proving knowledge of a Waters signature, we obtain a more efficient scheme by building the commitment algorithm on the verification equation of the underlying signature: recall that the verification equation of Waters signatures (σ_1, σ_2) returns 1 if and only if it holds that $e(\sigma_1, g) = e(g^\alpha, h) \cdot e(H_{\mathbb{G}}(M), \sigma_2)$, where $M \in \{0,1\}^L$ is the message and g^α, h are part of the public key. Now, to commit to a message $m \in \mathbb{Z}_p$ the sender can pick random $\theta_1, \theta_2 \in \mathbb{G}$ and compute $\mathsf{com} = e(g^\alpha, h)^m \cdot e(g, \theta_1) \cdot e(H_{\mathbb{G}}(\tau), \theta_2) \in \mathbb{G}_T$ and $\mathsf{dec} = (\theta_1, \theta_2)$. It is easy to see that a signature (σ_1, σ_2) on τ allows trapdoor opening com. Moreover, the resulting scheme gives shorter commitment string and a faster verification algorithm than in [24, 57].

Construction from the Strong Diffie-Hellman Assumption. As mentioned earlier, in the application to non-malleable commitments, simulation-sound trapdoor commitments only need to be secure against adversaries that choose beforehand (before receiving the public key) on which tags they will see equivocations of commitments produced by $\mathsf{FakeCom}$. In this case, we only need the underlying linearly homomorphic signature to be secure against non-adaptive Type I independent adversaries. The construction of Catalano, Fiore and Warinschi [27] is an example of such system. In [27], it was implicitly[8] proved that the scheme is secure against non-adaptive (independent) Type I adversaries under the strong Diffie-Hellman assumption [15].

Keygen(λ, n): given a security parameter $\lambda \in \mathbb{N}$ and an integer $n \in \mathsf{poly}(\lambda)$, choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$. Choose $\alpha \xleftarrow{R} \mathbb{Z}_p$, $g, v \xleftarrow{R} \mathbb{G}$ and $g_i \xleftarrow{R} \mathbb{G}$ for $i = 1$ to n. Finally, define the identifier space $\mathcal{T} := \mathbb{Z}_p$. The private key is $\mathsf{sk} := \alpha$ and the public key consists of

$$\mathsf{pk} := \Big((\mathbb{G}, \mathbb{G}_T),\ g,\ g^\alpha,\ v,\ \{g_i\}_{i=1}^n \Big).$$

Sign$(\mathsf{sk}, \tau, \vec{m})$: given a vector $\vec{m} = (m_1, \ldots, m_n) \in \mathbb{Z}_p^n$, a file identifier $\tau \in \mathbb{Z}_p$ and the private key $\mathsf{sk} = \alpha \in \mathbb{Z}_p$, choose $s \xleftarrow{R} \mathbb{Z}_p$. Then, compute a signature $\sigma = (\sigma_1, s) \in \mathbb{G} \times \mathbb{Z}_p$ where

$$\sigma_1 = \big(g_1^{m_1} \cdots g_n^{m_n} \cdot v^s \big)^{\frac{1}{\alpha + \tau}}.$$

SignDerive$(\mathsf{pk}, \tau, \{(\beta_i, \sigma_i)\}_{i=1}^\ell)$: given pk, a file identifier τ and ℓ tuples (β_i, σ_i), parse each signature σ_i as $\sigma_i = (\sigma_{i,1}, s_i)$ for $i = 1$ to ℓ. Then, compute

$$\sigma_1 = \prod_{i=1}^{\ell} \sigma_{i,1}^{\beta_i} \qquad\qquad s = \sum_{i=1}^{\ell} \beta_i \cdot s_i$$

[8] Catalano *et al.* [27] consider a model where the file identifiers are always chosen by the challenger at each signing query in the security game. However, the security proof of [27, Lemma 1] does not require the file identifiers to be uniformly distributed and it goes through if they are chosen by the adversary at the outset of the game instead of being chosen by the reduction.

and output (σ_1, s).

Verify($\mathsf{pk}, \tau, \vec{m}, \sigma$): given the public key pk, a signature $\sigma = (\sigma_1, s)$ and a message $(\tau, \vec{m})$, where $\tau \in \mathbb{Z}_p$ and $\vec{m} = (m_1, \ldots, m_n) \in (\mathbb{Z}_p)^n$, return 1 if and only if

$$e(\sigma_1, g^\tau \cdot g^\alpha) = e(g_1^{m_1} \cdots g_n^{m_n} \cdot v^s, g). \tag{20}$$

This construction can also be seen as a special case of our template where $\mathbb{G}_1 = \mathbb{G} \times \mathbb{Z}_p$ is a group for the operation $(\cdot, +)$ and $\mathbb{G}_2 = \mathbb{G}_T$ is a multiplicative group. Here, we thus have $|\mathbb{G}_1| = p^2$ and $|\mathbb{G}_2| = p$. The linear function F is now defined as

$$F\big((\sigma_1, s), \vec{m}, \mathsf{pk}, \tau\big) := e(\sigma_1, g^\tau \cdot g^\alpha) \cdot e(g_1^{m_1} \cdots g_n^{m_n} \cdot v^s, g^{-1}).$$

The linearly homomorphic signature of [27] thus implies a non-interactive non-adaptive simulation-sound trapdoor commitment to vectors based on the strong Diffie-Hellman assumption. Again, the scheme can be simplified by removing the term v^s since the underlying signature only needs to be secure against non-adaptive Type I attacks. In the case $n = 1$, the resulting non-malleable commitment is a variant of the one of [41, Section 4.2].

Non-Malleability from Malleability: Simulation-Sound Quasi-Adaptive NIZK Proofs and CCA2-Secure Encryption from Homomorphic Signatures

Benoît Libert[1] ⋆, Thomas Peters[2] ⋆⋆, Marc Joye[1], and Moti Yung[3]

[1] Ecole Normale Supérieure de Lyon, Laboratoire d'Informatique du Parallélisme (France)
[2] Technicolor (France)
[3] Université catholique de Louvain, Crypto Group (Belgium)
[4] Google Inc. and Columbia University (USA)

Abstract. Verifiability is central to building protocols and systems with integrity. Initially, efficient methods employed the Fiat-Shamir heuristics. Since 2008, the Groth-Sahai techniques have been the most efficient in constructing non-interactive witness indistinguishable and zero-knowledge proofs for algebraic relations in the standard model. For the important task of proving membership in linear subspaces, Jutla and Roy (Asiacrypt 2013) gave significantly more efficient proofs in the quasi-adaptive setting (QA-NIZK). For membership of the row space of a $t \times n$ matrix, their QA-NIZK proofs save $\Omega(t)$ group elements compared to Groth-Sahai. Here, we give QA-NIZK proofs made of a *constant* number group elements – regardless of the number of equations or the number of variables – and additionally prove them *unbounded* simulation-sound. Unlike previous unbounded simulation-sound Groth-Sahai-based proofs, our construction does not involve quadratic pairing product equations and does not rely on a chosen-ciphertext-secure encryption scheme. Instead, we build on structure-preserving signatures with homomorphic properties. We apply our methods to design new and improved CCA2-secure encryption schemes. In particular, we build the first efficient threshold CCA-secure keyed-homomorphic encryption scheme (*i.e.*, where homomorphic operations can only be carried out using a dedicated evaluation key) with publicly verifiable ciphertexts.

Keywords. NIZK proofs, simulation-soundness, chosen-ciphertext security, homomorphic cryptography.

1 Introduction

Non-interactive zero-knowledge proofs [8] play a fundamental role in the design of numerous cryptographic protocols. Unfortunately, until breakthrough results in the last decade [31–33], it was not known how to construct them efficiently without appealing to the random oracle methodology [7]. Groth and Sahai [33] described very efficient non-interactive witness indistinguishable (NIWI) and zero-knowledge (NIZK) proof systems for algebraic relations in groups equipped with a bilinear map. For these specific languages, the methodology of [33] does not require any proof of circuit satisfiability but rather leverages the properties of homomorphic commitments in bilinear groups. As a result, the length of each proof only depends on the number of equations and the number of variables.

While dramatically more efficient than general NIZK proofs, the GS techniques remain significantly more expensive than non-interactive proofs obtained from the Fiat-Shamir heuristic [26] in the random oracle model [7]: for example, proving that t variables satisfy a system of n linear equations demands $\Theta(t+n)$ group elements where Σ-protocols allow for $\Theta(t)$-size proofs. In addition, GS proofs are known to be malleable which, although useful in certain applications [5, 18], is undesirable when NIZK proofs serve as building blocks for non-malleable protocols. To construct chosen-ciphertext-secure encryption schemes [50], for example, the Naor-Yung/Sahai [46, 51] paradigm requires NIZK proofs satisfying a form of non-malleability called *simulation-soundness* [51]: informally, this property captures the inability of the adversary to prove false statements by itself, even after having observed simulated proofs for possibly false statements of its choice.

Groth-Sahai proofs can be made simulation-sound using constructions suggested in [32, 15, 34].

⋆ This work was done while this author was at Technicolor (France).

⋆⋆ This author was supported by the CAMUS Walloon Region Project.

However, even when starting from a linear equation, these techniques involve proofs for quadratic equations, which results in longer proofs. One-time simulation-soundness (*i.e.*, where the adversary only sees one simulated proof) is more economical to achieve as shown in [39, 42]. Jutla and Roy suggested a more efficient way to achieve a form of one-time simulation-soundness [37].

Quasi-Adaptive NIZK Proofs. For languages consisting of linear subspaces of a vector space, Jutla and Roy [38] recently showed how to significantly improve upon the efficiency of the GS paradigm in the *quasi-adaptive* setting. In quasi-adaptive NIZK proofs (QA-NIZK) for a class of languages $\{\mathcal{L}_\rho\}$ parametrized by ρ, the common reference string (CRS) is allowed to depend on the particular language $\mathcal{L}_\rho$ of which membership must be proved. At the same time, a single simulator should be effective for the whole class of languages $\{\mathcal{L}_\rho\}$. As pointed out in [38], QA-NIZK proofs are sufficient for many applications of Groth-Sahai proofs. In this setting, Jutla and Roy [38] gave very efficient QA-NIZK proofs of membership in linear subspaces. If $\mathbf{A} \in \mathbb{Z}_p^{t \times n}$ is a matrix or rank $t < n$, in order to prove membership of the language $\mathcal{L} = \{\boldsymbol{v} \in \mathbb{G}^n \mid \exists \boldsymbol{x} \in \mathbb{Z}_p^t \text{ s.t. } \boldsymbol{v} = g^{\boldsymbol{x} \cdot \mathbf{A}}\}$, the Jutla-Roy proofs only take $O(n-t)$ group elements – instead of $\Theta(n+t)$ in [33] – at the expense of settling for computational soundness. While highly efficient in the case $t \approx n$, these proofs remain of linear size in n and may result in long proofs when $t \ll n$, as is the case in, *e.g.*, certain applications of the Naor-Yung paradigm [15]. In the general case, we are still lacking a method for building proofs of size $O(t)$ – at least without relying on non-falsifiable assumptions [45] – which contrasts with the situation in the random oracle model.

The problem is even harder if we aim for simulation-soundness. While the Jutla-Roy solutions [38] nicely interact with their one-time simulation-sound proofs [37], they do not seem to readily extend into unbounded simulation-sound (USS) proofs (where the adversary can see an arbitrary number of simulated proofs before outputting a proof of its own) while retaining the same efficiency. For this reason, although they can be applied in specific cases like [15], we cannot always use them in a modular way to build CCA2-secure encryption schemes in scenarios where security definitions involve many challenge ciphertexts.

Our Contributions. Recently, in [43], it was pointed out that structure-preserving signatures (SPS) [3, 2] with (additive) homomorphic properties have unexpected applications in the design of non-malleable structure-preserving commitments. Here, we greatly extend their range of applications and demonstrate that they can surprisingly be used (albeit non-generically) in the design of strongly non-malleable primitives like simulation-sound proofs and chosen-ciphertext-secure cryptosystems.

Concretely, we describe unbounded simulation-sound QA-NIZK proofs of *constant-size* for linear subspaces. The length of a proof does not depend on the number of equations or the number of variables, but only on the underlying assumption. Like those of [38], our proofs are computationally sound under standard assumptions[5]. Somewhat surprisingly, they are even asymptotically shorter than random-oracle-based proofs derived from Σ-protocols.

Moreover, our construction provides *unbounded* simulation-soundness. Under the Decision Linear assumption [10], we obtain QA-NIZK arguments consisting of 15 group elements and a one-time signature with its verification key. As it turns out, it is also the first unbounded simulation-sound proof system that does not involve quadratic pairing product equations or a CCA2-secure encryption scheme. Efficiency comparisons (given in Appendix E) show that we only need 20 group elements per proof where the best USS extension [15] of Groth-Sahai costs $6t + 2n + 52$ group elements. Under the k-linear assumption, the proof length becomes $O(k^2)$ and thus avoids any dependency on the subspace dimension. Our proof system builds on the linearly homomorphic structure-preserving signatures of Libert, Peters, Joye and Yung [43], which allow signing vectors of group elements without knowing their discrete logarithms.

[5] Note that these results do not contradict the impossibility results of Gentry and Wichs [30] because, in the quasi-adaptive setting, the CRS may hide a trapdoor that allows recognizing elements of the language. The proof of [30] applies to reductions that cannot efficiently detect when the adversary breaks the soundness property.

For applications, like CCA2 security [46, 51], where only one-time simulation-soundness is needed, we further optimize our proof system and obtain a relatively simulation-sound QA-NIZK proof system, as defined in [37], with constant-size proofs. Under the DLIN assumption (resp. the k-linear assumption), we achieve relative simulation-soundness with only 4 (resp. $k + 2$) group elements!

As the first application of USS proofs, we construct a chosen-ciphertext-secure keyed-homomorphic encryption scheme with threshold decryption. Keyed-homomorphic encryption is a primitive, suggested by Emura *et al.* [24], where homomorphic ciphertext manipulations are only possible to a party holding a devoted evaluation key SK_h which, by itself, does not enable decryption. The scheme should provide IND-CCA2 security when the evaluation key is unavailable to the adversary and remain IND-CCA1 secure when SK_h is exposed. Other approaches to reconcile homomorphism and non-malleability were taken in [47–49, 12, 18] but they inevitably satisfy weaker security notions than adaptive chosen-ciphertext security [50]. The results of [24] showed that CCA2-security does not rule out homomorphicity when the capability to compute over encrypted data is restricted.

Emura *et al.* [24] gave realizations of chosen-ciphertext-secure keyed-homomorphic schemes based on hash proof systems [21]. However, these do not readily enable threshold decryption – as would be desirable in voting protocols – since valid ciphertexts are not publicly recognizable, which makes it harder to prove CCA security in the threshold setting. Moreover, these solutions are not known to satisfy the strongest security definition of [24]. The reason is that this definition seemingly requires a form of unbounded simulation-soundness. Our QA-NIZK proofs fulfill this requirement and provide an efficient CCA2-secure threshold keyed-homomorphic system where ciphertexts are 65% shorter than in instantiations of the same high-level idea using previous simulation-sound proofs.

Using our relatively simulation-sound QA-NIZK proofs, we then build adaptively secure non-interactive threshold cryptosystems with CCA2 security and improved efficiency. The constructions of Libert and Yung [42] were improved by Escala *et al.* [25]. So far, the most efficient solution is obtained from the Jutla-Roy results [37, 38] via relatively sound proofs [37]. Using our relatively sound QA-NIZK proof system, we shorten ciphertexts by $\Theta(k)$ elements under the k-linear assumption.

Our Techniques. In our unbounded simulation-sound proofs, each QA-NIZK proof can be seen as a Groth-Sahai NIWI proof of knowledge of a one-time linearly homomorphic signature on the vector that allegedly belongs to the linear subspace. Here, the NIWI proof is generated for a Groth-Sahai CRS that depends on the verification key of a one-time signature (following an idea of Malkin *et al.* [44]), the private key of which is used to sign the entire proof so as to prevent re-randomizations. The reason why it provides unbounded simulation-soundness is that, with non-negligible probability, the CRS is perfectly hiding on all simulated proofs and extractable in the adversarially-generated fake proof. Hence, if the adversary manages to prove membership of a vector outside the linear subspace, the reduction is able to extract a homomorphic signature that it would not have been able to compute itself, thereby breaking the DLIN assumption. At a high level, the system can be seen as a two-tier proof system made of a non-malleable proof of knowledge of a malleable proof of membership.

In our optimized relatively-sound proofs, we adapt ideas of Jutla and Roy [37] and combine the one-time linearly homomorphic signature of [43] with a smooth-projective hash function [21].

Our threshold keyed-homomorphic cryptosystem combines a hash proof system and a publicly verifiable USS proof that the ciphertext is well-formed. The keyed-homomorphic property is achieved by using the simulation trapdoor of the proof system as an evaluation key SK_h, allowing the evaluator to generate proofs without knowing the witnesses. As implicitly done in [24] in the case of hash proof systems, the simulation trapdoor is thus used in the scheme and not only in the security proof.

2 Background and Definitions

2.1 Quasi-Adaptive NIZK Proofs

Quasi-Adaptive NIZK (QA-NIZK) proofs are NIZK proofs where the CRS is allowed to depend on the specific language for which proofs have to be generated. The CRS is divided into a fixed part Γ, produced by an algorithm K_0, and a language-dependent part ψ. However, there should be a single simulator for the entire class of languages.

Let λ be a security parameter. For public parameters Γ produced by K_0, let $\mathcal{D}_\Gamma$ be a probability distribution over a collection of relations $\mathcal{R} = \{R_\rho\}$ parametrized by a string ρ with an associated language $\mathcal{L}_\rho = \{x \mid \exists w : R_\rho(x, w) = 1\}$.

We consider proof systems where the prover and the verifier both take a label lbl as additional input. For example, this label can be the message-carrying part of an Elgamal-like encryption. Formally, a tuple of algorithms $(\mathsf{K}_0, \mathsf{K}_1, \mathsf{P}, \mathsf{V})$ is a QA-NIZK proof system for $\mathcal{R}$ if there exists a PPT simulator $(\mathsf{S}_1, \mathsf{S}_2)$ such that, for any PPT adversaries $\mathcal{A}_1, \mathcal{A}_2$ and $\mathcal{A}_3$, we have the following properties:

Quasi-Adaptive Completeness:

$$\begin{aligned}&\Pr[\Gamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ \psi \leftarrow \mathsf{K}_1(\Gamma, \rho);\\ &\qquad (x, w, \mathsf{lbl}) \leftarrow \mathcal{A}_1(\Gamma, \psi, \rho);\ \pi \leftarrow \mathsf{P}(\psi, x, w, \mathsf{lbl}) : \mathsf{V}(\psi, x, \pi, \mathsf{lbl}) = 1 \ \text{ if } R_\rho(x, w) = 1] = 1.\end{aligned}$$

Quasi-Adaptive Soundness:

$$\begin{aligned}&\Pr[\Gamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ \psi \leftarrow \mathsf{K}_1(\Gamma, \rho);\ (x, \pi, \mathsf{lbl}) \leftarrow \mathcal{A}_2(\Gamma, \psi, \rho) :\\ &\qquad\qquad \mathsf{V}(\psi, x, \pi, \mathsf{lbl}) = 1 \ \wedge\ \neg(\exists w : R_\rho(x, w) = 1)] \in \mathsf{negl}(\lambda).\end{aligned}$$

Quasi-Adaptive Zero-Knowledge:

$$\begin{aligned}&\Pr[\Gamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ \psi \leftarrow \mathsf{K}_1(\Gamma, \rho)\ :\ \mathcal{A}_3^{\mathsf{P}(\psi,.,.,.)}(\Gamma, \psi, \rho) = 1]\\ &\qquad \approx \Pr[\Gamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ (\psi, \tau_{sim}) \leftarrow \mathsf{S}_1(\Gamma, \rho)\ :\ \mathcal{A}_3^{\mathsf{S}(\psi,\tau_{sim},.,.,.)}(\Gamma, \psi, \rho) = 1],\end{aligned}$$

where

- $\mathsf{P}(\psi, ., ., .)$ emulates the actual prover. It takes as input (x, w) and lbl and outputs a proof π if $(x, w) \in R_\rho$. Otherwise, it outputs $\perp$.
- $\mathsf{S}(\psi, \tau_{sim}, ., ., .)$ is an oracle that takes as input (x, w) and lbl. It outputs a simulated proof $\mathsf{S}_2(\psi, \tau_{sim}, x, \mathsf{lbl})$ if $(x, w) \in R_\rho$ and $\perp$ if $(x, w) \notin R_\rho$.

We assume that the CRS ψ contains an encoding of ρ, which is thus available to V. The definition of Quasi-Adaptive Zero-Knowledge requires a single simulator for the entire family of relations $\mathcal{R}$.

2.2 Simulation-Soundness and Relative Soundness

It is often useful to have a property called *simulation-soundness*, which requires that the adversary be unable to prove false statements even after having seen simulated proofs for possibly false statements.

Unbounded Simulation-Soundness: For any PPT adversary $\mathcal{A}_4$, it holds that

$$\begin{aligned}&\Pr[\Gamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ (\psi, \tau_{sim}) \leftarrow \mathsf{S}_1(\Gamma, \rho);\ (x, \pi, \mathsf{lbl}) \leftarrow \mathcal{A}_4^{\mathsf{S}_2(\psi,\tau_{sim},.,.)}(\Gamma, \psi, \rho)\ :\\ &\qquad \mathsf{V}(\psi, x, \pi, \mathsf{lbl}) = 1\ \wedge\ \neg(\exists w : R_\rho(x, w) = 1)\ \wedge\ (x, \pi, \mathsf{lbl}) \notin Q] \in \mathsf{negl}(\lambda),\end{aligned}$$

where the adversary is allowed unbounded access to an oracle $\mathsf{S}_2(\psi, \tau, ., .)$ that takes as input statement-label pairs (x, lbl) (where x may be outside $\mathcal{L}_\rho$) and outputs simulated proofs $\pi \leftarrow \mathsf{S}_2(\psi, \tau_{sim}, x, \mathsf{lbl})$ before updating the set $Q = Q \cup \{(x, \pi, \mathsf{lbl})\}$, which is initially empty.

In the weaker notion of one-time simulation-soundness, only one query to the S_2 oracle is allowed.

In some applications, one may settle for a weaker notion, called *relative soundness* by Jutla and Roy [37], which allows for more efficient proofs, especially in the single-theorem case. Informally, relatively sound proof systems involve both a public verifier *and* a private verification algorithm, which has access to a trapdoor. For hard languages, the two verifiers should almost always agree on any adversarially-created proof. Moreover, the private verifier should not accept a non-trivial proof for a false statement, even if the adversary has already seen proofs for false statements.

A labeled single-theorem relatively sound QA-NIZK proof system is comprised of a quasi-adaptive labeled proof system $(\mathsf{K}_0, \mathsf{K}_1, \mathsf{P}, \mathsf{V})$ along with an efficient private verifier W and an efficient simulator $(\mathsf{S}_1, \mathsf{S}_2)$. Moreover, the following properties should hold for any PPT adversaries $(\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4)$.

Quasi Adaptive Relative Single-Theorem Zero-Knowledge:

$$\begin{aligned} &\Pr[\varGamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow D_\varGamma;\ \psi \leftarrow \mathsf{K}_1(\varGamma, \rho);\ (x, w, \mathsf{lbl}, s) \leftarrow \mathcal{A}_1^{\mathsf{V}(\psi,\cdot,\cdot)}(\varGamma, \psi, \rho); \\ &\qquad\qquad \pi \leftarrow \mathsf{P}(\psi, \rho, x, w, \mathsf{lbl}) : \mathcal{A}_2^{\mathsf{V}(\psi,\cdot,\cdot)}(\pi, s) = 1] \\ &\approx \Pr[\varGamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow D_\varGamma;\ (\psi, \tau) \leftarrow \mathsf{S}_1(\varGamma, \rho);\ (x, w, \mathsf{lbl}, s) \leftarrow \mathcal{A}_1^{\mathsf{W}(\psi,\tau,\cdot,\cdot)}(\varGamma, \psi, \rho); \\ &\qquad\qquad \pi \leftarrow \mathsf{S}_2(\psi, \rho, \tau, x, \mathsf{lbl}) : \mathcal{A}_2^{\mathsf{W}(\psi,\tau,\cdot,\cdot)}(\pi, s) = 1], \end{aligned}$$

Here, $\mathcal{A}_1$ is restricted to choosing (x, w) such that $R_\rho(x, w) = 1$.

Quasi Adaptive Relative Single-Theorem Simulation-Soundness:

$$\begin{aligned} &\Pr[\varGamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow D_\varGamma;\ (\psi, \tau) \leftarrow \mathsf{S}_1(\varGamma, \rho);\ (x, \mathsf{lbl}, s) \leftarrow \mathcal{A}_3^{\mathsf{W}(\psi,\tau,\cdot,\cdot,\cdot)}(\varGamma, \psi, \rho); \\ &\qquad \pi \leftarrow \mathsf{S}_2(\psi, \rho, \tau, x, \mathsf{lbl}) : (x', \mathsf{lbl}', \pi') \leftarrow \mathcal{A}_4^{\mathsf{W}(\psi,\tau,\cdot,\cdot,\cdot)}(s, \pi) : \\ &\qquad (x, \pi, \mathsf{lbl}) \neq (x', \pi', \mathsf{lbl}') \ \wedge \ \nexists w' \text{ s.t. } R_\rho(x', w') = 1 \ \wedge \ \mathsf{W}(\psi, \tau, x', \mathsf{lbl}', \pi') = 1] \in \mathsf{negl}(\lambda) \end{aligned}$$

Note that the definition of relative simulation-soundness does not require the adversary to provide a witness but the definition of single-theorem zero-knowledge does.

2.3 Definitions for Threshold Keyed-Homomorphic Encryption

A (t, N)-threshold keyed-homomorphic encryption scheme consists of the following algorithms.

Keygen(λ, t, N)**:** takes as input a security parameter λ and integers $t, N \in \mathsf{poly}(\lambda)$ (with $1 \leq t \leq N$), where N is the number of decryption servers and $t \leq N$ is the decryption threshold. It outputs $(PK, SK_h, \mathbf{VK}, \mathbf{SK}_d)$, where PK is the public key, SK_h is the homomorphic evaluation key, $\mathbf{SK}_d = (SK_{d,1}, \ldots, SK_{d,N})$ is a vector of private key shares and $\mathbf{VK} = (VK_1, \ldots, VK_N)$ is a vector of verification keys. For each i, the decryption server i is given the share $(i, SK_{d,i})$. The verification key VK_i will be used to check the validity of decryption shares generated using $SK_{d,i}$.

Encrypt(PK, M)**:** takes a input a public key PK and a plaintext M. It outputs a ciphertext C.

Ciphertext-Verify(PK, C)**:** takes as input a public key PK and a ciphertext C. It outputs 1 if C is deemed valid w.r.t. PK and 0 otherwise.

Share-Decrypt$(PK, i, SK_{d,i}, C)$**:** on input of a public key PK, a ciphertext C and a private-key share $(i, SK_{d,i})$, this (possibly randomized) algorithm outputs a special symbol $(i, \perp)$ if **Ciphertext-Verify**$(PK, C) = 0$. Otherwise, it outputs a decryption share $\mu_i = (i, \hat{\mu}_i)$.

Share-Verify(PK, VK_i, C, μ_i)**:** takes in PK, the verification key VK_i, a ciphertext C and a purported decryption share $\mu_i = (i, \hat{\mu}_i)$. It outputs either 1 or 0. In the former case, μ_i is said to be a *valid* decryption share. We adopt the convention that $(i, \perp)$ is an invalid decryption share.

Combine$(PK, \mathbf{VK}, C, \{\mu_i\}_{i \in S})$**:** takes in $(PK, \mathbf{VK}, C)$ and a t-subset $S \subset \{1, \ldots, N\}$ with decryption shares $\{\mu_i\}_{i \in S}$. It outputs either a plaintext M or $\perp$ if $\{\mu_i\}_{i \in S}$ contains invalid shares.

Eval$(PK, SK_h, C^{(1)}, C^{(2)})$**:** takes as input PK, the evaluation key SK_h and ciphertexts $C^{(1)}$, $C^{(2)}$. If **Ciphertext-Verify**$(PK, C^{(j)}) = 0$ for some $j \in \{1, 2\}$, the algorithm returns $\perp$. Otherwise, it conducts a binary homomorphic operation over $C^{(1)}$ and $C^{(2)}$ and outputs a ciphertext C.

The above syntax assumes a trusted dealer. It generalizes that of ordinary threshold cryptosystems. By setting $SK_h = \varepsilon$ and discarding the evaluation algorithm, we obtain a classical threshold system.

Definition 1. *A threshold keyed-homomorphic public-key cryptosystem is secure against chosen-ciphertext attacks (or KH-CCA secure) if no PPT adversary has noticeable advantage in this game:*

1. *The challenger runs* $\mathbf{Keygen}(\lambda)$ *to obtain a public key* PK*, vectors* $\mathbf{SK}_d = (SK_{d,1}, \ldots, SK_{d,N})$, $\mathbf{VK} = (VK_1, \ldots, VK_N)$ *and a homomorphic evaluation key* SK_h*. It gives* PK *and* $\mathbf{VK}$ *to the adversary* $\mathcal{A}$ *and keeps* $(SK_h, \mathbf{SK}_d)$ *to itself. In addition, the challenge initializes a set* $\mathcal{D} \leftarrow \emptyset$*, which is initially empty.*

2 *The adversary* $\mathcal{A}$ *adaptively makes queries to the following oracles on multiple occasions:*

   - *Corruption query: at any time,* $\mathcal{A}$ *may decide to corrupt a decryption server. To this end, it specifies an index* $i \in \{1, \ldots, N\}$ *and obtains the private key share* $SK_{d,i}$*.*
   - *Evaluation query:* $\mathcal{A}$ *can invoke the evaluation oracle* $\mathsf{Eval}(SK_h, .)$ *on a pair* $(C^{(1)}, C^{(2)})$ *of ciphertexts of its choice. If there exists* $j \in \{1, 2\}$ *such that* $\mathbf{Ciphertext}\text{-}\mathbf{Verify}(PK, C^{(j)}) = 0$*, return* $\perp$*. Otherwise, the oracle* $\mathsf{Eval}(SK_h, .)$ *computes* $C \leftarrow \mathbf{Eval}(SK_h, C^{(1)}, C^{(2)})$ *and returns* C*. In addition, if* $C^{(1)} \in \mathcal{D}$ *or* $C^{(2)} \in \mathcal{D}$*, it sets* $\mathcal{D} \leftarrow \mathcal{D} \cup \{C\}$*.*
   - *Reveal query: at any time,* $\mathcal{A}$ *may also decide to corrupt the evaluator by invoking the* RevHK *oracle on a unique occasion. The oracle responds by returning* SK_h*.*
   - *Decryption query:* $\mathcal{A}$ *can also invoke the partial decryption oracle on arbitrary ciphertexts* C *and indexes* $i \in \{1, \ldots, n\}$*. If* $\mathbf{Ciphertext}\text{-}\mathbf{Verify}(PK, C) = 0$ *or if* $C \in \mathcal{D}$*, the oracle returns* $\perp$*. Otherwise, the oracle returns the decryption share* $\mu_i \leftarrow \mathbf{Share}\text{-}\mathbf{Decrypt}(PK, i, SK_{d,i}, C)$*.*

3. *The adversary* $\mathcal{A}$ *chooses two equal-length messages* M_0, M_1 *and obtains* $C^\star = \mathbf{Encrypt}(PK, M_\beta)$ *for some random bit* $\beta \xleftarrow{R} \{0, 1\}$*. In addition, the challenger sets* $\mathcal{D} \leftarrow \mathcal{D} \cup \{C^\star\}$*.*
4. $\mathcal{A}$ *makes further queries as in step 2 with some restrictions. Namely,* $\mathcal{A}$ *cannot corrupt more than* $t - 1$ *servers throughout the entire game. Moreover, if* $\mathcal{A}$ *chooses to obtain* SK_h *(via the* RevHK *oracle) at some point, no more post-challenge decryption query is allowed beyond that point.*
5. $\mathcal{A}$ *outputs a bit* β' *and is deemed successful if* $\beta' = \beta$*. As usual,* $\mathcal{A}$*'s advantage is measured as the distance* $\mathbf{Adv}(\mathcal{A}) = |\Pr[\beta' = \beta] - \frac{1}{2}|$*.*

It is important to note that, even if $\mathcal{A}$ chooses to obtain SK_h immediately after having seen the public key PK, it still has access to the decryption oracle *before* the challenge phase. In other words, the scheme should remain IND-CCA1 if $\mathcal{A}$ is given PK and SK_h at the outset of the game. After the challenge phase, decryption queries are allowed until the moment when the adversary obtains SK_h.

In [24], Emura *et al.* suggested a weaker definition where the adversary is not allowed to query the evaluation oracle on derivatives of the challenge ciphertext. As a consequence, the set $\mathcal{D}$ is always the singleton $\{C^\star\}$ after step 3. In this paper, we will stick to the stronger definition.

2.4 Hardness Assumptions

For simplicity, we use symmetric bilinear maps $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ over groups of prime order p, but extensions to the asymmetric setting $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$ are possible.

Definition 2 ([10]). *The* **Decision Linear Problem** *(DLIN) in* $\mathbb{G}$*, is to distinguish the distributions* $(g^a, g^b, g^{ac}, g^{bd}, g^{c+d})$ *and* $(g^a, g^b, g^{ac}, g^{bd}, g^z)$*, with* $a, b, c, d \xleftarrow{R} \mathbb{Z}_p$*,* $z \xleftarrow{R} \mathbb{Z}_p$*.*

We sometimes use the Simultaneous Double Pairing (SDP) assumption, which is weaker than DLIN. As noted in [17], any algorithm solving SDP immediately yields a DLIN distinguisher.

Definition 3. *The* **Simultaneous Double Pairing problem** *(SDP) in* $(\mathbb{G}, \mathbb{G}_T)$ *is, given group elements* $(g_z, g_r, h_z, h_u) \in \mathbb{G}^4$*, to find a non-trivial triple* $(z, r, u) \in \mathbb{G}^3 \backslash \{(1_\mathbb{G}, 1_\mathbb{G}, 1_\mathbb{G})\}$ *such that* $e(g_z, z) \cdot e(g_r, r) = 1_{\mathbb{G}_T}$ *and* $e(h_z, z) \cdot e(h_u, u) = 1_{\mathbb{G}_T}$*.*

2.5 Linearly Homomorphic Structure-Preserving Signatures

Linearly homomorphic SPS schemes are homomorphic signatures where messages and signatures live in the domain group $\mathbb{G}$ (see Appendix B for syntactic definitions) of a bilinear map. Libert *et al.* [43] described the following one-time construction and proved its security under the SDP assumption. By "one-time", we mean that only one linear subspace can be safely signed using a given key pair.

Keygen(λ, n): given a security parameter λ and the dimension $n \in \mathbb{N}$ of vectors to be signed, choose bilinear group $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$. Choose $g_z, g_r, h_z, h_u \xleftarrow{R} \mathbb{G}$. Then, for $i = 1$ to n, pick $\chi_i, \gamma_i, \delta_i \xleftarrow{R} \mathbb{Z}_p$ and compute $g_i = {g_z}^{\chi_i} {g_r}^{\gamma_i}$ and $h_i = {h_z}^{\chi_i} {h_u}^{\delta_i}$. The private key is $\mathsf{sk} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^n$ while the public key is $\mathsf{pk} = \big(g_z, g_r, h_z, h_u, \{(g_i, h_i)\}_{i=1}^n\big)$.

Sign$(\mathsf{sk}, (M_1, \ldots, M_n))$: to sign a vector $(M_1, \ldots, M_n) \in \mathbb{G}^n$ using $\mathsf{sk} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^n$, compute and return $(z, r, u) = \big(\prod_{i=1}^n M_i^{-\chi_i},\ \prod_{i=1}^n M_i^{-\gamma_i},\ \prod_{i=1}^n M_i^{-\delta_i}\big) \in \mathbb{G}^3$.

SignDerive$(\mathsf{pk}, \{(\omega_i, \sigma^{(i)})\}_{i=1}^\ell)$: given a public key pk and ℓ tuples $(\omega_i, \sigma^{(i)})$, where $\omega_i \in \mathbb{Z}_p$ for each i, parse $\sigma^{(i)}$ as $\sigma^{(i)} = \big(z_i, r_i, u_i\big) \in \mathbb{G}^3$ for $i = 1$ to ℓ. Then, compute and return $\sigma = (z, r, u)$, where $z = \prod_{i=1}^\ell z_i^{\omega_i}$, $r = \prod_{i=1}^\ell r_i^{\omega_i}$ and $u = \prod_{i=1}^\ell u_i^{\omega_i}$.

Verify$(\mathsf{pk}, \sigma, (M_1, \ldots, M_n))$: given a signature $\sigma = (z, r, u) \in \mathbb{G}^3$ and a vector $(M_1, \ldots, M_n)$, return 1 if and only if $(M_1, \ldots, M_n) \neq (1_\mathbb{G}, \ldots, 1_\mathbb{G})$ and (z, r, u) satisfy

$$1_{\mathbb{G}_T} = e(g_z, z) \cdot e(g_r, r) \cdot \prod_{i=1}^n e(g_i, M_i), \qquad 1_{\mathbb{G}_T} = e(h_z, z) \cdot e(h_u, u) \cdot \prod_{i=1}^n e(h_i, M_i). \tag{1}$$

One particularity of this scheme is that, even if the private key is available, it is difficult to find two distinct signatures on the same vector if the SDP assumption holds: by dividing out the two signatures, one obtains the solution of an SDP instance (g_z, g_r, h_z, h_u) contained in the public key.

Two constructions of full-fledged (as opposed to one-time) linearly homomorphic SPS were given in [43]. One of these will serve as a basis for our proof system and is recalled in Appendix C. In these constructions, all algorithms additionally input a tag which identifies the dataset that vectors belongs to. Importantly, only vectors associated with the same tag can be homomorphically combined.

3 Unbounded Simulation-Sound Quasi-Adaptive NIZK Arguments

In the following, vectors are always considered as row vectors unless stated otherwise. If $\mathbf{A} \in \mathbb{Z}_p^{t \times n}$ is a matrix, we denote by $g^\mathbf{A} \in \mathbb{G}^{t \times n}$ the matrix obtained by exponentiating g using the entries of $\mathbf{A}$.

We consider public parameters $\mathit{\Gamma} = (\mathbb{G}, \mathbb{G}_T, g)$ consisting of bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ with a generator $g \in \mathbb{G}$. Like [38], we will consider languages $\mathcal{L}_\rho = \{g^{\boldsymbol{x} \cdot \mathbf{A}} \in \mathbb{G}^n \mid \boldsymbol{x} \in \mathbb{Z}_p^t\}$ that are parametrized by $\rho = g^\mathbf{A} \in \mathbb{G}^{t \times n}$, where $\mathbf{A} \in \mathbb{Z}_q^{t \times n}$ is a $t \times n$ matrix of rank $t < n$.

As in [38], we assume that the distribution $\mathcal{D}_\mathit{\Gamma}$ is efficiently samplable: there exists a PPT algorithm which outputs a pair $(\rho, \mathbf{A})$ describing a relation R_ρ and its associated language $\mathcal{L}_\rho$ according to $\mathcal{D}_\mathit{\Gamma}$. One example of such a distribution is obtained by picking a uniform matrix $\mathbf{A} \xleftarrow{R} \mathbb{Z}_p^{t \times n}$ – which has full rank with overwhelming probability – and setting $\rho = g^\mathbf{A}$.

Our construction builds on the homomorphic signature recalled in Appendix C. Specifically, the language-dependent CRS ψ contains one-time linearly homomorphic signatures on the rows of the matrix $\rho \in \mathbb{G}^{t \times n}$. For each vector $\boldsymbol{v} \in \mathcal{L}_\rho$, the prover can use the witness $\boldsymbol{x} \in \mathbb{Z}_p^t$ to derive and prove knowledge of a one-time homomorphic signature (z, r, u) on $\boldsymbol{v}$. This signature (z, r, u) is already a QA-NIZK proof of membership but it does not provide simulation-soundness. To acquire this property, we follow [44] and generate a NIWI proof of knowledge of (z, r, u) for a Groth-Sahai CRS that depends on the verification key of an ordinary one-time signature. The latter's private key is used to sign the NIWI proof so as to prevent unwanted proof manipulations. Using the private key of the homomorphic one-time signature as a trapdoor, the simulator is also able to create proofs for vectors

$\boldsymbol{v} \notin \mathcal{L}_\rho$. Due to the use of perfectly NIWI proofs, these fake proofs do not leak any more information about the simulation key than the CRS does. At the same time, the CRS can be prepared in such a way that, with non-negligible probability, it becomes perfectly binding on an adversarially-generated proof, which allows extracting a non-trivial signature on a vector $\boldsymbol{v} \notin \mathcal{L}_\rho$.

Like [38], our quasi-adaptive NIZK proof system $(\mathsf{K}_0, \mathsf{K}_1, \mathsf{P}, \mathsf{V})$ is a split CRS construction in that K_1 can be divided into two algorithms $(\mathsf{K}_{10}, \mathsf{K}_{11})$. The first one K_{10} outputs some state information s and a first CRS $\mathbf{CRS}_2$ which is only used by the verifier and does not depend on the language $\mathcal{L}_\rho$. The second part K_{11} of K_1 inputs the state information s and the output of Γ of K_0 and outputs $\mathbf{CRS}_1$ which is only used by the prover. The construction goes as follows.

$\mathsf{K}_0(\lambda)$: choose groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ with $g \stackrel{R}{\leftarrow} \mathbb{G}$. Then, output $\Gamma = (\mathbb{G}, \mathbb{G}_T, g)$

The dimensions (t, n) of the matrix $\mathbf{A} \in \mathbb{Z}_p^{t \times n}$ can be either fixed or part of the language, so that t, n can be given as input to the CRS generation algorithm K_1.

$\mathsf{K}_1(\Gamma, \rho)$: parse Γ as $(\mathbb{G}, \mathbb{G}_T, g)$ and ρ as a matrix $\rho = \big(G_{i,j}\big)_{1 \leq i \leq t,\ 1 \leq j \leq n} \in \mathbb{G}^{t \times n}$.

1. Generate a key pair $(\mathsf{pk}_{rand}, \mathsf{sk}_{rand})$ for the randomizable signature of Appendix C in order to sign vectors of $\mathbb{G}^n$. Namely, choose $g_z, g_r, h_z, h_u \stackrel{R}{\leftarrow} \mathbb{G}$ and do the following.
   a. For $i = 1$ to n, pick $\chi_i, \gamma_i, \delta_i \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and compute $g_i = {g_z}^{\chi_i} {g_r}^{\gamma_i}$ and $h_i = {h_z}^{\chi_i} {h_u}^{\delta_i}$.
   b. Generate $L + 1$ Groth-Sahai common reference strings, for some $L \in \mathsf{poly}(\lambda)$. To this end, choose $f_1, f_2 \stackrel{R}{\leftarrow} \mathbb{G}$ and define $\boldsymbol{f}_1 = (f_1, 1, g) \in \mathbb{G}^3$, $\boldsymbol{f}_2 = (1, f_2, g) \in \mathbb{G}^3$. Then, pick $\boldsymbol{f}_{3,i} \stackrel{R}{\leftarrow} \mathbb{G}^3$ for $i = 0$ to L.

   Let $\mathsf{sk}_{rand} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^n$ be the private key and the matching public key is

$$\mathsf{pk}_{rand} = \Big(\ g_z,\ g_r,\ h_z,\ h_u,\ \{(g_i, h_i)\}_{i=1}^n,\ \mathbf{f} = \big(\boldsymbol{f}_1, \boldsymbol{f}_2, \{\boldsymbol{f}_{3,i}\}_{i=0}^L\big)\ \Big).$$

2. Use sk_{rand} to generate one-time linearly homomorphic signatures $\{(z_i, r_i, u_i)\}_{i=1}^t$ on the vectors $\boldsymbol{\rho}_i = (G_{i1}, \ldots, G_{in}) \in \mathbb{G}^n$ that form the rows of ρ. These are obtained as

$$(z_i, r_i, u_i) = \big(\prod_{j=1}^n G_{i,j}^{-\chi_j},\ \prod_{j=1}^n G_{i,j}^{-\gamma_j},\ \prod_{j=1}^n G_{i,j}^{-\delta_j}\big) \qquad \forall i \in \{1, \ldots, t\}.$$

3. Choose a strongly unforgeable one-time signature $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ with verification keys consisting of L-bit strings.
4. The CRS $\psi = (\mathbf{CRS}_1, \mathbf{CRS}_2)$ consists of two parts which are defined as

$$\mathbf{CRS}_1 = \Big(\rho,\ \mathsf{pk}_{rand},\ \{(z_i, r_i, u_i)\}_{i=1}^t,\ \Sigma\Big), \qquad \mathbf{CRS}_2 = \Big(\mathsf{pk}_{rand},\ \Sigma\Big),$$

   while the simulation trapdoor τ_{sim} is $\mathsf{sk}_{rand} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^n$.

$\mathsf{P}(\Gamma, \psi, \boldsymbol{v}, x, \mathsf{lbl})$: given a candidate $\boldsymbol{v} \in \mathbb{G}^n$ and a witness $\boldsymbol{x} = (x_1, \ldots, x_t) \in \mathbb{Z}_p^t$ such that $\boldsymbol{v} = g^{\boldsymbol{x} \cdot \mathbf{A}}$, generate a one-time signature key pair $(\mathsf{SVK}, \mathsf{SSK}) \leftarrow \mathcal{G}(\lambda)$ and do the following.

1. Using $\{(z_j, r_j, u_j)\}_{j=1}^t$, derive a one-time linearly homomorphic signature (z, r, u) on $\boldsymbol{v}$. Namely, compute $z = \prod_{i=1}^t z_i^{x_i}$, $r = \prod_{i=1}^t r_i^{x_i}$ and $u = \prod_{i=1}^t u_i^{x_i}$.
2. Using $\mathsf{SVK} = \mathsf{SVK}[1] \ldots \mathsf{SVK}[L] \in \{0,1\}^L$, define the vector $\boldsymbol{f}_{\mathsf{SVK}} = \boldsymbol{f}_{3,0} \cdot \prod_{i=1}^L \boldsymbol{f}_{3,i}^{\mathsf{SVK}[i]}$ and assemble a Groth-Sahai CRS $\mathbf{f}_{\mathsf{SVK}} = (\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_{\mathsf{SVK}})$. Using $\mathbf{f}_{\mathsf{SVK}}$, generate commitments $\boldsymbol{C}_z$, $\boldsymbol{C}_r$, $\boldsymbol{C}_u$ to the components of $(z, r, u) \in \mathbb{G}^3$ along with NIWI proofs $(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$ that $\boldsymbol{v}$ and (z, r, u) satisfy (1). Let $(\boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2) \in \mathbb{G}^{15}$ be the resulting commitments and proofs.
3. Generate $\sigma = \mathcal{S}(\mathsf{SSK}, (\boldsymbol{v}, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \mathsf{lbl}))$ and output

$$\pi = (\mathsf{SVK}, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \sigma) \tag{2}$$

$\mathsf{V}(\Gamma, \psi, \boldsymbol{v}, \pi, \mathsf{lbl})$**:** parse π as per (2) and return 1 if (i) $\mathcal{V}(\mathsf{SVK}, (\boldsymbol{v}, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \mathsf{lbl}), \sigma) = 1$; (ii) $(\boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$ forms a valid NIWI proof for the CRS $\mathbf{f}_{\mathsf{SVK}} = (\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_{\mathsf{SVK}})$ (see (4) in Appendix C for the detailed verification equations). If either condition fails to hold, return 0.

In order to simulate a proof for a given vector $\boldsymbol{v} \in \mathbb{G}^n$, the simulator uses $\tau_{sim} = \mathsf{sk}_{rand}$ to generate a fresh one-time homomorphic signature on $\boldsymbol{v} \in \mathbb{G}^n$ and proceeds as in steps 2-3 of algorithm P.

The proof π only consists of 15 group elements and a one-time pair (SVK, σ). Remarkably, its length does not depend on the number of equations n or the number of variables t. In comparison, Groth-Sahai proofs already require $3t + 2n$ group elements in their basic form and become even more expensive when it comes to achieve unbounded simulation-soundness. The Jutla-Roy techniques [38] reduce the proof length to $2(n-t)$ elements – which only competes with our proofs when $t \approx n$ – but it is unclear how to extend them to get unbounded simulation-soundness without affecting their efficiency. Our CRS consists of $O(t+n+L)$ group elements against $O(t(n-t))$ in [38]. More detailed comparisons are given in Appendix E between proof systems based on the DLIN assumption.

Interestingly, the above scheme even outperforms Fiat-Shamir-like proofs derived from Σ-protocols which would give $\Theta(t)$-size proofs here. The construction readily extends to rely on the k-linear assumption for $k > 2$. In this case, the proof comprises $(k+1)(2k+1)$ elements and its size thus only depends on k, as detailed in Appendix D.

Moreover, the verification algorithm only involves *linear* pairing product equations whereas all known unbounded simulation-sound extensions of Groth-Sahai proofs require either quadratic equations or a linearization step involving extra variables.

We finally remark that, if we give up the simulation-soundness property, the proof length drops to $k+1$ group elements under the k-linear assumption.

Theorem 1. *The scheme is an unbounded simulation-sound QA-NIZK proof system if the DLIN assumption holds in $\mathbb{G}$ and Σ is strongly unforgeable.* (The proof is given in Appendix F).

We note that the above construction is not tightly secure as the gap between the simulation-soundness adversary's advantage and the probability to break the DLIN assumption depends on the number of simulated proofs obtained by the adversary. For applications like tightly secure public-key encryption [34], it would be interesting to modify the proof system to obtain tight security.

4 Single-Theorem Relatively Sound Quasi-Adaptive NIZK Arguments

In applications where single-theorem relatively sound NIZK proofs suffice, we can further improve the efficiency. Under the k-linear assumption, the proof length reduces from $O(k^2)$ elements to $O(k)$ elements. Under the DLIN assumption, each proof fits within 4 elements and only costs $2n+6$ pairings to verify. In comparison, the verifier needs $2(n-t)(t+2)$ pairing evaluations in [38].

As in [37], we achieve relative soundness using smooth projective hash functions [21]. To this end, we need to encode the matrix $\rho \in \mathbb{G}^{t \times n}$ as a $2t \times (2n+1)$ matrix.

$\mathsf{K}_0(\lambda)$**:** choose groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ with $g \overset{R}{\leftarrow} \mathbb{G}$. Then, output $\Gamma = (\mathbb{G}, \mathbb{G}_T, g)$.

Again, the dimensions of $\mathbf{A} \in \mathbb{Z}_p^{t \times n}$ can be either fixed or part of $\mathcal{L}_\rho$, so that t, n can be given as input to the CRS generation algorithm K_1.

$\mathsf{K}_1(\Gamma, \rho)$**:** parse Γ as $(\mathbb{G}, \mathbb{G}_T, g)$ and ρ as $\rho = \big(G_{ij}\big)_{1 \le i \le t,\ 1 \le j \le n} \in \mathbb{G}^{t \times n}$ and do the following.

1. Choose two n-vectors $\boldsymbol{d} = (d_1, \ldots, d_n) \overset{R}{\leftarrow} \mathbb{Z}_p^n$ and $\boldsymbol{e} = (e_1, \ldots, e_n) \overset{R}{\leftarrow} \mathbb{Z}_p^n$ in order to define $\boldsymbol{W} = (W_1, \ldots, W_t) = g^{\mathbf{A} \cdot \boldsymbol{d}^\top} \in \mathbb{G}^t$ and $\boldsymbol{Y} = (Y_1, \ldots, Y_t) = g^{\mathbf{A} \cdot \boldsymbol{e}^\top} \in \mathbb{G}^t$. These will be used to define a projective hash function.

2. Generate a key pair $(\mathsf{pk}_{ots}, \mathsf{sk}_{ots})$ for the one-time linearly homomorphic signature of Section 2.5 in order to sign vectors in $\mathbb{G}^{2n+1}$. Let $\mathsf{pk}_{ots} = \big((\mathbb{G}, \mathbb{G}_T), g_z, g_r, h_z, h_u, \{(g_i, h_i)\}_{i=1}^{2n+1}\big)$ be the public key and let $\mathsf{sk}_{ots} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^{2n+1}$ be the corresponding private key.
3. Use sk_{ots} to generate one-time linearly homomorphic signatures $\{(z_i, r_i, u_i)\}_{i=1}^{2t}$ on the independent vectors below, which are obtained from the rows of the matrix $\rho = \big(G_{i,j}\big)_{1 \leq i \leq t,\ 1 \leq j \leq n}$.

$$\begin{aligned} \boldsymbol{H}_{2i-1} &= (G_{i,1}, \ldots, G_{i,n}, Y_i,\ 1\ ,\ \ \ldots\ \ ,1\ ) \in \mathbb{G}^{2n+1} \qquad\qquad i \in \{1, \ldots, t\} \\ \boldsymbol{H}_{2i} &= (1\ ,\ \ \ldots\ \ ,1\ , W_i, G_{i,1}, \ldots, G_{i,n}) \in \mathbb{G}^{2n+1} \end{aligned}$$

4. Choose a collision-resistant hash function $H : \{0,1\}^* \to \mathbb{Z}_p$.
5. The CRS ψ consists of a first part $\mathbf{CRS}_1$ that is only used by the prover and a second part $\mathbf{CRS}_2$ which is only used by the verifier. These are defined as

$$\mathbf{CRS}_1 = \Big(\rho,\ \mathsf{pk}_{ots},\ \boldsymbol{W},\ \boldsymbol{Y},\ \{(z_i, r_i, u_i)\}_{i=1}^{2t},\ H\Big), \qquad \mathbf{CRS}_2 = \Big(\mathsf{pk}_{ots},\ \boldsymbol{W},\ \boldsymbol{Y},\ H\Big).$$

The simulation trapdoor τ_{sim} is sk_{ots} and the private verification trapdoor is $\tau_v = \{\boldsymbol{d}, \boldsymbol{e}\}$.

$\mathsf{P}(\mathit{\Gamma}, \psi, \boldsymbol{v}, x, \mathsf{lbl})$: given a candidate $\boldsymbol{v} \in \mathbb{G}^n$, a witness $\boldsymbol{x} = (x_1, \ldots, x_t) \in \mathbb{Z}_p^t$ such that $\boldsymbol{v} = g^{\boldsymbol{x} \cdot \mathbf{A}}$ and a label lbl, compute $\alpha = H(\rho, \boldsymbol{v}, \mathsf{lbl}) \in \mathbb{Z}_p$. Then, using $\{(z_i, r_i, u_i)\}_{i=1}^{2t}$, derive a one-time linearly homomorphic signature (z, r, u) on the vector $\tilde{\boldsymbol{v}} = \big(v_1, \ldots, v_n, \pi_0, v_1^\alpha, \ldots, v_n^\alpha\big) \in \mathbb{G}^{2n+1}$, where $\pi_0 = \prod_{i=1}^t (W_i^\alpha Y_i)^{x_i}$. Namely, compute and output the proof $\pi = (z, r, u, \pi_0) \in \mathbb{G}^4$, where

$$z = \prod_{i=1}^t (z_{2i-1} \cdot z_{2i}^\alpha)^{x_i}, \qquad r = \prod_{i=1}^t (r_{2i-1} \cdot r_{2i}^\alpha)^{x_i}, \qquad u = \prod_{i=1}^t (u_{2i-1} \cdot u_{2i}^\alpha)^{x_i} \qquad \pi_0 = \prod_{i=1}^t (W_i^\alpha Y_i)^{x_i}$$

$\mathsf{V}(\mathit{\Gamma}, \psi, \boldsymbol{v}, \pi, \mathsf{lbl})$: parse $\boldsymbol{v}$ as $(v_1, \ldots, v_n) \in \mathbb{G}^n$ and π as $(z, r, u, \pi_0) \in \mathbb{G}^4$. Compute $\alpha = H(\rho, \boldsymbol{v}, \mathsf{lbl})$ and return 1 if and only if (z, r, u) is a valid signature on $\tilde{\boldsymbol{v}} = (v_1, \ldots, v_n, \pi_0, v_1^\alpha, \ldots, v_n^\alpha) \in \mathbb{G}^{2n+1}$. Namely, it should satisfy the equalities $1_{\mathbb{G}_T} = e(g_z, z) \cdot e(g_r, r) \cdot \prod_{i=1}^n e(g_i \cdot g_{i+n+1}^\alpha, v_i) \cdot e(g_{n+1}, \pi_0)$ and $1_{\mathbb{G}_T} = e(h_z, z) \cdot e(h_u, u) \cdot \prod_{i=1}^n e(h_i \cdot h_{i+n+1}^\alpha, v_i) \cdot e(h_{n+1}, \pi_0)$.

$\mathsf{W}(\mathit{\Gamma}, \psi, \tau_v, \boldsymbol{v}, \pi, \mathsf{lbl})$: given $\boldsymbol{v} = (v_1, \ldots, v_n) \in \mathbb{G}^n$, parse π as $(z, r, u, \pi_0) \in \mathbb{G}^4$ and τ_v as $\{\boldsymbol{d}, \boldsymbol{e}\}$, with $\boldsymbol{d} = (d_1, \ldots, d_n) \in \mathbb{Z}_p^n$ and $\boldsymbol{e} = (e_1, \ldots, e_n) \in \mathbb{Z}_p^n$. Compute $\alpha = H(\rho, \boldsymbol{v}, \mathsf{lbl}) \in \mathbb{Z}_p$ and return 0 if the public verification test V fails. Otherwise, return 1 if $\pi_0 = \prod_{j=1}^n v_j^{e_j + \alpha d_j}$ and 0 otherwise.

We note that, while the proving algorithm is deterministic, each statement has many valid proofs. However, finding two valid proofs for the same statement is computationally hard, as will be shown in the proof of Theorem 2.

The scheme readily extends to rest on the k-linear assumption with $k > 2$. In this case, the proof requires $k + 2$ group elements – whereas combining the techniques of [37, 38] demands $k(n + 1 - t)$ elements per proof – and a CRS of size $O(k(n+t))$. From a security standpoint, we prove the following result in Appendix G.

Theorem 2. *The above proof system is a relatively sound QA-NIZK proof system if the SDP assumption holds in $(\mathbb{G}, \mathbb{G}_T)$ and if H is a collision-resistant hash function.*

As an application, we describe a new adaptively secure CCA2-secure non-interactive threshold cryptosystem based on the DLIN assumption in Appendix I. Under the k-linear assumption, the scheme provides ciphertexts that are $\Theta(k)$ group elements shorter than in previous such constructions. Under the DLIN assumption, ciphertexts consist of 8 elements of $\mathbb{G}$, which spares one group element w.r.t. the best previous variants [37, 38] of Cramer-Shoup with publicly verifiable ciphertexts.

5 An Efficient Threshold Keyed-Homomorphic KH-CCA-Secure Encryption Scheme from the DLIN Assumption

The use of linearly homomorphic signatures as publicly verifiable proofs of ciphertext validity in the Cramer-Shoup paradigm [20, 21] was suggested in [43]. However, the latter work only discusses non-adaptive (*i.e.*, CCA1) attacks. In the CCA2 case, a natural idea is to proceed as in our unbounded simulation-sound proof system and use the verification key of a on-time signature as the tag of a homomorphic signature: since cross-tag homomorphic operations are disallowed, the one-time signature will prevent illegal ciphertext manipulations after the challenge phase.

To obtain the desired keyed-homomorphic property, we use the simulation trapdoor of a simulation-sound proof system as the homomorphic evaluation key. This approach was already used by Emura *et al.* [24] in the context of designated verifier proofs. Here, publicly verifiable proofs are obtained from a homomorphic signature scheme of which the private key serves as an evaluation key: anyone equipped with this key can multiply two ciphertexts (or, more precisely, their built-in homomorphic components), generate a new tag and sign the resulting ciphertext using the private key of the homomorphic signature. Moreover, we can leverage the fact that the latter private key is always available to the reduction in the security proof of the homomorphic signature [43]. In the game of Definition 1, the simulator can thus hand over the evaluation key SK_h to the adversary upon request.

Emura *et al.* [24] gave constructions of KH-CCA secure encryption schemes based on hash proof systems [21]. However, these constructions are only known to provide a relaxed flavor of KH-CCA security where evaluation queries should not involve derivatives of the challenge ciphertext. The reason is that 2-universal hash proof systems [21] only provide a form of one-time simulation soundness whereas the model of Definition 1 seemingly requires unbounded simulation-soundness. Indeed, when the evaluation oracle is queried on input of a derivative of the challenge ciphertext in the security proof, the homomorphic operation may result in a ciphertext containing a vector outside the language $\mathcal{L}_\rho$. Since the oracle has to simulate a proof for this vector, each homomorphic evaluation can carry a proof for a potentially false statement. In some sense, each output of the evaluation oracle can be seen as yet another challenge ciphertext. In this setting, our efficient unbounded simulation-sound QA-NIZK proof system comes in handy.

It remains to make sure that CCA1 security is always preserved, should the adversary obtain the evaluation key SK_h at the outset of the game. To this end, we include a second derived one-time homomorphic signature (Z, R, U) in the ciphertext without including its private key in SK_h.

Keygen(λ, t, N)**:** Choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$. Then, do the following.

1. Pick $f, g, h \overset{R}{\leftarrow} \mathbb{G}$, $x_0, x_1, x_2 \overset{R}{\leftarrow} \mathbb{Z}_p$ and set $X_1 = f^{x_1} g^{x_0} \in \mathbb{G}$, $X_2 = h^{x_2} g^{x_0} \in \mathbb{G}$. Then, define $\boldsymbol{f} = (f, 1, g) \in \mathbb{G}^3$ and $\boldsymbol{h} = (1, h, g) \in \mathbb{G}^3$.
2. Choose random polynomials $P_1[Z], P_2[Z], P[Z] \in \mathbb{Z}_p[Z]$ of degree $t-1$ such that $P_1(0) = x_1$, $P_2(0) = x_2$ and $P(0) = x_0$. For each $i \in \{1, \ldots, N\}$, compute $VK_i = (Y_{i,1}, Y_{i,2})$ where $Y_{i,1} = f^{P_1(i)} g^{P(i)}$ and $Y_{i,2} = h^{P_2(i)} g^{P(i)}$.
3. Choose $f_{r,1}, f_{r,2} \overset{R}{\leftarrow} \mathbb{G}$ in order to define vectors $\boldsymbol{f}_{r,1} = (f_{r,1}, 1, g)$, $\boldsymbol{f}_{r,2} = (1, f_{r,2}, g)$ and $\boldsymbol{f}_{r,3} = \boldsymbol{f}_{r,1}^{\phi_1} \cdot \boldsymbol{f}_{r,2}^{\phi_2} \cdot (1, 1, g)^{-1}$, where $\phi_1, \phi_2 \overset{R}{\leftarrow} \mathbb{Z}_p$. These vectors will be used as a Groth-Sahai CRS for the generation of NIZK proofs showing the validity of decryption shares.
4. Choose a strongly unforgeable one-time signature $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ with verification keys consisting of L-bit strings, for some $L \in \mathsf{poly}(\lambda)$.
5. Generate a key pair for the one-time linearly homomorphic structure-preserving signature of Section 2.5 with $n = 3$. Let $\mathsf{pk}_{ot} = \big(G_z, G_r, H_z, H_u, \{(G_i, H_i)\}_{i=1}^3\big)$ be the public key and let $\mathsf{sk}_{ot} = \{(\varphi_i, \vartheta_i, \varpi_i)\}_{i=1}^3$ be the corresponding private key.
6. Generate one-time homomorphic signatures $\{(Z_j, R_j, U_j)\}_{j=1,2}$ on the vectors $\boldsymbol{f} = (f, 1, g)$ and $\boldsymbol{h} = (1, h, g)$. These signatures consist of $(Z_1, R_1, U_1) = \big(f^{-\varphi_1} g^{-\varphi_3}, f^{-\vartheta_1} g^{-\vartheta_3}, f^{-\varpi_1} g^{-\varpi_3}\big)$ and $(Z_2, R_2, U_2) = \big(h^{-\varphi_2} g^{-\varphi_3}, h^{-\vartheta_2} g^{-\vartheta_3}, h^{-\varpi_2} g^{-\varpi_3}\big)$ and erase sk_{ot}.

7. Generate a key pair $(\mathsf{pk}_{rand}, \mathsf{sk}_{rand})$ as in step 1 of the proof system in Section 3 with $n = 3$. Let $\mathsf{sk}_{rand} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^{3}$ be the private key for which the corresponding public key is

$$\mathsf{pk}_{rand} = \Big(g_z,\ g_r,\ h_z,\ h_u,\ \{(g_i, h_i)\}_{i=1}^{3},\ \mathbf{f} = \big(\boldsymbol{f}_1, \boldsymbol{f}_2, \{\boldsymbol{f}_{3,i}\}_{i=0}^{L}\big) \Big).$$

8. Use sk_{rand} to generate one-time linearly homomorphic signatures $\{(z_j, r_j, u_j)\}_{j=1,2}$ on the independent vectors $\boldsymbol{f} = (f, 1, g) \in \mathbb{G}^3$ and $\boldsymbol{h} = (1, h, g) \in \mathbb{G}^3$. These are obtained as

$$(z_1, r_1, u_1) = \big(f^{-\chi_1} g^{-\chi_3}, f^{-\gamma_1} g^{-\gamma_3}, f^{-\delta_1} g^{-\delta_3}\big), \qquad (z_2, r_2, u_2) = \big(h^{-\chi_2} g^{-\chi_3}, h^{-\gamma_2} g^{-\gamma_3}, h^{-\delta_2} g^{-\delta_3}\big).$$

9. The public key is defined to be

$$PK = \Big(g,\ \boldsymbol{f},\ \boldsymbol{h},\ \boldsymbol{f}_{r,1},\ \boldsymbol{f}_{r,2},\ \boldsymbol{f}_{r,3},\ X_1,\ X_2,\ \mathsf{pk}_{ot},\ \mathsf{pk}_{rand},\ \{(Z_j, R_j, U_j)\}_{j=1}^{2}\ \{(z_j, r_j, u_j)\}_{j=1}^{2}\Big).$$

The evaluation key is $SK_h = \mathsf{sk}_{rand} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^{3}$ while the i-th decryption key share is defined to be $SK_{d,i} = (P_1(i), P_2(i), P(i))$. The vector of verification keys is defined as $\mathbf{VK} = (VK_1, \ldots, VK_N)$, where $VK_i = (Y_{i,1}, Y_{i,2})$ for $i = 1$ to N.

Encrypt(M, PK)**:** to encrypt $M \in \mathbb{G}$, generate a one-time signature key pair $(\mathsf{SVK}, \mathsf{SSK}) \leftarrow \mathcal{G}(\lambda)$.

1. Choose $\theta_1, \theta_2 \overset{R}{\leftarrow} \mathbb{Z}_p$ and compute

$$C_0 = M \cdot X_1^{\theta_1} \cdot X_2^{\theta_2}, \qquad C_1 = f^{\theta_1}, \qquad C_2 = h^{\theta_2}, \qquad C_3 = g^{\theta_1 + \theta_2}.$$

2. Construct a first linearly homomorphic signature (Z, R, U) on the vector $(C_1, C_2, C_3) \in \mathbb{G}^3$. Namely, compute $Z = Z_1^{\theta_1} \cdot Z_2^{\theta_2}$, $R = R_1^{\theta_1} \cdot R_2^{\theta_2}$ and $U = U_1^{\theta_1} \cdot U_2^{\theta_2}$.
3. Using $\{(z_j, r_j, u_j)\}_{j=1,2}$, derive another homomorphic signature (z, r, u) on (C_1, C_2, C_3). Namely, compute $z = z_1^{\theta_1} \cdot z_2^{\theta_2}$, $r = r_1^{\theta_1} \cdot r_2^{\theta_2}$ and $u = u_1^{\theta_1} \cdot u_2^{\theta_2}$.
4. Using $\mathsf{SVK} = \mathsf{SVK}[1] \ldots \mathsf{SVK}[L] \in \{0,1\}^L$, define the vector $\boldsymbol{f}_{\mathsf{SVK}} = \boldsymbol{f}_{3,0} \cdot \prod_{i=1}^{L} \boldsymbol{f}_{3,i}^{\mathsf{SVK}[i]}$ and assemble a Groth-Sahai CRS $\mathbf{f}_{\mathsf{SVK}} = (\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_{\mathsf{SVK}})$. Using $\mathbf{f}_{\mathsf{SVK}}$, generate commitments $\boldsymbol{C}_z$, $\boldsymbol{C}_r$, $\boldsymbol{C}_u$ to the components of $(z, r, u) \in \mathbb{G}^3$ along with proofs $(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$ as in step 2 of the proving algorithm of Section 3. Let $(\boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2) \in \mathbb{G}^{15}$ be the resulting NIWI proof.
5. Generate $\sigma = \mathcal{S}(\mathsf{SSK}, (C_0, C_1, C_2, C_3, Z, R, U, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2))$ and output

$$C = (\mathsf{SVK}, C_0, C_1, C_2, C_3, Z, R, U, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \sigma) \tag{3}$$

Ciphertext-Verify$\big(PK, C\big)$**:** parse C as in (3). Return 1 if and only if these conditions are satisfied: (i) $\mathcal{V}(\mathsf{SVK}, (C_0, C_1, C_2, C_3, Z, R, U, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2), \sigma) = 1$; (ii) $(Z, R, U) \in \mathbb{G}^3$ is a valid homomorphic signature on (C_1, C_2, C_3); (iii) $(\boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2) \in \mathbb{G}^{15}$ is a valid proof w.r.t. the CRS $(\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_{\mathsf{SVK}})$ that committed (z, r, u) satisfy the relations (1) for the vector $(C_1, C_2, C_3) \in \mathbb{G}^3$. Here, we define $\boldsymbol{f}_{\mathsf{SVK}} = \boldsymbol{f}_{3,0} \cdot \prod_{i=1}^{L} \boldsymbol{f}_{3,i}^{\mathsf{SVK}[i]}$.

Share-Decrypt$(PK, i, SK_{d,i}, C)$**:** on inputs $SK_{d,i} = (P_1(i), P_2(i), P(i)) \in \mathbb{Z}_p^3$ and C, return $(i, \perp)$ if **Ciphertext-Verify**$\big(PK, C\big) = 0$. Otherwise, compute $\hat{\mu}_i = \big(\nu_i, \boldsymbol{C}_{P_1}, \boldsymbol{C}_{P_2}, \boldsymbol{C}_P, \pi_{\mu_i}\big)$ which consists of a partial decryption $\nu_i = C_1^{P_1(i)} \cdot C_2^{P_2(i)} \cdot C_3^{P(i)}$ as well as commitments $\boldsymbol{C}_{P_1}, \boldsymbol{C}_{P_2}, \boldsymbol{C}_P$ to exponents $P_1(i), P_2(i), P(i) \in \mathbb{Z}_p$ and a proof π_{ν_i} that these satisfy the equations

$$\nu_i = C_1^{P_1(i)} \cdot C_2^{P_2(i)} \cdot C_3^{P(i)}, \qquad Y_{i,1} = f^{P_1(i)} g^{P(i)}, \qquad Y_{i,2} = h^{P_2(i)} g^{P(i)}.$$

The commitments $\boldsymbol{C}_{P_1}, \boldsymbol{C}_{P_2}, \boldsymbol{C}_P$ and the proof π_{ν_i} are generated using the CRS $(\boldsymbol{f}_{r,1}, \boldsymbol{f}_{r,2}, \boldsymbol{f}_{r,3})$ (see Appendix A for details). Then, return $\mu_i = (i, \hat{\mu}_i)$.

Share-Verify$\big(PK, VK_i, C, (i, \hat{\mu_i})\big)$**:** parse C as in (3) and VK_i as $(Y_{i,1}, Y_{i,2})$. If $\hat{\mu_i} = \perp$ or $\hat{\mu_i}$ cannot be parsed as $\big(\nu_i, \boldsymbol{C}_{P_1}, \boldsymbol{C}_{P_2}, \boldsymbol{C}_P, \pi_{\mu_i}\big)$, return 0. Otherwise, return 1 if and only if π_{μ_i} is valid.

Combine$(PK, \mathbf{VK}, C, \{(i, \hat{\mu_i})\}_{i \in S})$**:** for each $i \in S$, parse the share $\hat{\mu_i}$ as $\big(\nu_i, \boldsymbol{C}_{P_1}, \boldsymbol{C}_{P_2}, \boldsymbol{C}_P, \pi_{\mu_i}\big)$ and return $\perp$ if **Share-Verify**$\big(PK, C, (i, \hat{\mu_i})\big) = 0$. Otherwise, compute $\nu = \prod_{i \in S} \nu_i^{\Delta_{i,S}(0)}$, which equals $\nu = C_1^{x_1} \cdot C_2^{x_2} \cdot C_3^{x_0} = X_1^{\theta_1} \cdot X_2^{\theta_2}$ and in turn reveals $M = C_0/\nu$.

Eval$(PK, SK_h, C^{(1)}, C^{(2)})$**:** parse SK_h as $\{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^{3}$. For each $j \in \{1, 2\}$, parse $C^{(j)}$ as

$$C^{(j)} = (\mathsf{SVK}^{(j)}, C_0^{(j)}, C_1^{(j)}, C_2^{(j)}, C_3^{(j)}, Z^{(j)}, R^{(j)}, U^{(j)}, \boldsymbol{C}_z^{(j)}, \boldsymbol{C}_r^{(j)}, \boldsymbol{C}_u^{(j)}, \boldsymbol{\pi}_1^{(j)}, \boldsymbol{\pi}_2^{(j)}, \sigma^{(j)})$$

and return $\perp$ if either $C^{(1)}$ or $C^{(2)}$ is invalid. Otherwise,

1. Compute $C_0 = \prod_{j=1}^{2} C_0^{(j)}$, $C_1 = \prod_{j=1}^{2} C_1^{(j)}$, $C_2 = \prod_{j=1}^{2} C_2^{(j)}$ and $C_3 = \prod_{j=1}^{2} C_3^{(j)}$ as well as $Z = \prod_{j=1}^{2} Z^{(j)}$, $R = \prod_{j=1}^{2} R^{(j)}$ and $U = \prod_{j=1}^{2} U^{(j)}$.
2. Generate a new one-time signature key pair $(\mathsf{SVK}, \mathsf{SSK}) \leftarrow \mathcal{G}(\lambda)$. Using $SK_h = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^{3}$, generate proof elements $\boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2$ on the vector (C_1, C_2, C_3) using the simulator of the proof system in Section 3 with the one-time verification key SVK.
3. Return the derived ciphertext $C = (\mathsf{SVK}, C_0, C_1, C_2, C_3, Z, R, U, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \sigma)$ where $\sigma = \mathcal{S}(\mathsf{SSK}, (C_0, C_1, C_2, C_3, Z, R, U, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2))$.

In Appendix H, we prove the KH-CCA security of the scheme assuming that Σ is a strongly unforgeable one-time signature and that the DLIN assumption holds in $\mathbb{G}$.

In some applications, it may be desirable to add an extra randomization step to the evaluation algorithm in order to make sure that derived ciphertexts will be indistinguishable from freshly generated encryption (similarly to [48]). It is straightforward to modify the scheme to obtain this property.

If the scheme is instantiated using Groth's one-time signature [32], the ciphertext consists of 25 elements of $\mathbb{G}$ and two elements of $\mathbb{Z}_p$. It is interesting to compare the above system with an instantiation of the same design principle using the best known Groth-Sahai-based unbounded simulation-sound proof [15][Appendix A.2], which requires 65 group elements in this specific case. With this proof system, we end up with 77 group elements per ciphertexts under the DLIN assumption (assuming that an element of $\mathbb{Z}_p$ has the same length as the representation of a group element). The above realization thus saves 50 group elements and compresses ciphertexts to 35% of their original length.

We note that it is possible to adapt the scheme to rely on the Symmetric eXternal Diffie-Hellman assumption in asymmetric pairings $e : \mathbb{G} \times \hat{\mathbb{G}} \rightarrow \mathbb{G}_T$. In this case, the ciphertext contains 9 elements of $\mathbb{G}$, 2 elements of $\hat{\mathbb{G}}$ and a one-time key pair (SVK, σ). Using the one-time signature of [32], the ciphertext overhead amounts to 4096 bits on Barreto-Naehrig curves [4] if each element of $\mathbb{G}$ (resp. each element of $\hat{\mathbb{G}}$) has a 256-bit (resp. 512-bit) representation.

References

1. M. Abe, S. Fehr. Adaptively Secure Feldman VSS and Applications to Universally-Composable Threshold Cryptography. In *Crypto'04*, *LNCS* 3152, pp. 317–334, 2004.
2. M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, M. Ohkubo. Structure-Preserving Signatures and Commitments to Group Elements. In *Crypto'10*, *LNCS* 6223, pp. 209–236, 2010.
3. M.Abe, K. Haralambiev, M. Ohkubo. Signing on Elements in Bilinear Groups for Modular Protocol Design. In Cryptology ePrint Archive: Report 2010/133, 2010.
4. P. Barreto, M. Naehrig. Pairing-Friendly Elliptic Curves of Prime Order. In *SAC'05*, *LNCS* 3897, pp. 319–331, 2005.
5. M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya, H. Shacham. Randomizable Proofs and Delegatable Anonymous Credentials. In *Crypto'09*, *LNCS* 5677, pp. 108–125, 2009.
6. M. Bellare, T. Ristenpart. Simulation without the Artificial Abort: Simplified Proof and Improved Concrete Security for Waters' IBE Scheme. In *Eurocrypt'09*, *LNCS* 5479, pp. 407–424, 2009.
7. M. Bellare, P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS*, pp. 62–73, 1993.
8. M. Blum, P. Feldman, S. Micali. Non-Interactive Zero-Knowledge and Its Applications. In *STOC'88*, pp. 103-112, 1988.

9. D. Boneh, X. Boyen, S. Halevi. Chosen Ciphertext Secure Public Key Threshold Encryption Without Random Oracles. In *CT-RSA'06*, *LNCS* 3860, pp. 226–243, 2006.
10. D. Boneh, X. Boyen, H. Shacham. Short group signatures. In *Crypto'04*, LNCS 3152, pp. 41–55, 2004.
11. D. Boneh, M. Hamburg, S. Halevi, R. Ostrovsky. Circular-Secure Encryption from Decision Diffie-Hellman. In *Crypto'08*, LNCS 5157, pp. 108–125, 2008.
12. D. Boneh, G. Segev, B. Waters. Targeted malleability: homomorphic encryption for restricted computations. In *ITCS 2012*, pp. 350–366, 2012.
13. C. Boyd. Digital Multisignatures. In *Cryptography and Coding*, Oxford University Press, pp. 241–246, 1989.
14. X. Boyen, Q. Mei, B. Waters. Direct Chosen Ciphertext Security from Identity-Based Techniques. in *ACM CCS'05*, pp. 320–329, 2005.
15. J. Camenisch, N. Chandran, and V. Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In *Eurocrypt'09*, LNCS 5479, pp. 351–368, 2009.
16. R. Canetti, R. Gennaro, S. Jarecki, H. Krawczyk, T. Rabin. Adaptive Security for Threshold Cryptosystems. In *Crypto'99*, *LNCS* 1666, pp. 98–115, 1999.
17. J. Cathalo, B. Libert, M. Yung. Group Encryption: Non-Interactive Realization in the Standard Model. In *Asiacrypt'09*, *LNCS* 5912, pp. 179–196, 2009.
18. M. Chase, M. Kohlweiss, A. Lysyanskaya, S. Meiklejohn. Malleable Proof Systems and Applications. In *Eurocrypt'12*, *LNCS* 7237, pp. 281–300, 2012.
19. J.-S. Coron, T. Lepoint, M. Tibouchi. Practical Multilinear Maps over the Integers. In *Crypto'13*, *LNCS* 8042, pp. 476–493, 2013.
20. R. Cramer, V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Crypto'98*, LNCS 1462, pp. 13–25, 1998.
21. R. Cramer, V. Shoup. Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In *Eurocrypt'02*, *LNCS* 2332, pp. 45–64, 2002.
22. Y. Desmedt. Society and Group Oriented Cryptography: A New Concept. In *Crypto'87*, *LNCS* 293, pp. 120–127, 1987.
23. Y. Desmedt, Y. Frankel. Threshold Cryptosystems. In *Crypto'89*, *LNCS* 435, pp. 307–315, 1989.
24. K. Emura, G. Hanaoka, G. Ohtake, T. Matsuda, S. Yamada. Chosen Ciphertext Secure Keyed-Homomorphic Public-Key Encryption. In *PKC'13*, *LNCS* 7778, pp. 32–50, 2013.
25. A. Escala, G. Herold, E. Kiltz, C. Ràfols, J. Villar. An Algebraic Framework for Diffie-Hellman Assumptions. In *Crypto'13*, *LNCS* 8043, pp. 129–147, 2013.
26. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Crypto'86*, LNCS 263, pages 186–194, 1986.
27. P.-A. Fouque, D. Pointcheval. Threshold Cryptosystems Secure against Chosen-Ciphertext Attacks. In *Asiacrypt'01*, *LNCS* 2248, pp. 351–368, 2001.
28. Y. Frankel, P. MacKenzie, M. Yung. Adaptively-Secure Distributed Public-Key Systems. In *ESA'99*, *LNCS* 1643, pp. 4–27, 1999.
29. S. Garg, C. Gentry, S. Halevi. Candidate Multilinear Maps from Ideal Lattices. In *Eurocrypt'13*, *LNCS* 7881, pp. 1–17, 2013.
30. C. Gentry, D. Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In *STOC'11*, pp. 99-108, 2011.
31. J. Groth, R. Ostrovsky, and A. Sahai. Perfect non-interactive zero knowledge for NP. In *Eurocrypt'06*, *LNCS* 4004, pp. 339–358, 2006.
32. J. Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In *Asiacrypt'06*, *LNCS* 4284, pp. 444–459, 2006.
33. J. Groth, A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Eurocrypt'08*, LNCS 4965, pp. 415–432, 2008.
34. D. Hofheinz, T. Jager. Tightly Secure Signatures and Public-Key Encryption. In *Crypto'12*, *LNCS* 7417, pp. 590–607, 2012.
35. D. Hofheinz, E. Kiltz. Programmable Hash Functions and Their Applications. In *Crypto'08*, *LNCS* 5157, pp. 21–38, 2008.
36. S. Jarecki, A. Lysyanskaya. Adaptively Secure Threshold Cryptography: Introducing Concurrency, Removing Erasures. In *Eurocrypt'00*, *LNCS* 1807, pp. 221–242, 2000.
37. C. Jutla, A. Roy. Relatively-Sound NIZKs and Password-Based Key-Exchange. In *PKC'12*, *LNCS* 7293, pp. 485–503, 2012.
38. C. Jutla, A. Roy. Shorter Quasi-Adaptive NIZK Proofs for Linear Subspaces. In *Asiacrypt'13*, *LNCS* series, 2013. Cryptology ePrint Archive: Report 2013/109, 2013.
39. J. Katz, V. Vaikuntanathan. Round-Optimal Password-Based Authenticated Key Exchange. In *TCC'11*, LNCS 6597, pp. 293–310, 2011.
40. A. Lewko. Tools for Simulating Features of Composite Order Bilinear Groups in the Prime Order Setting. In *Eurocrypt'12*, *LNCS* 7237, pp. 318–335, 2012.
41. B. Libert, M. Yung. Adaptively Secure Non-Interactive Threshold Cryptosystems. In *ICALP 2011*, LNCS 6756, pp. 588–600, 2011.

42. B. Libert, M. Yung. Non-Interactive CCA2-Secure Threshold Cryptosystems with Adaptive Security: New Framework and Constructions. In *TCC 2012*, LNCS 7194, pp. 75–93, 2012.
43. B. Libert, T. Peters, M. Joye, M. Yung. Linearly Homomorphic Structure-Preserving Signatures and their Applications. In *Crypto 2013*, *LNCS* 8043, pp. 289–307, 2013.
44. T. Malkin, I. Teranishi, Y. Vahlis, M. Yung. Signatures resilient to continual leakage on memory and computation. In *TCC'11*, LNCS 6597, pp. 89–106, 2011.
45. M. Naor. On cryptographic assumptions and challenges. In *Crypto'03*, *LNCS* 2729, pp. 96–109, 2003.
46. M. Naor, M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC'90*, pp. 427–437, 1990.
47. M. Prabhakaran, M. Rosulek. Rerandomizable RCCA Encryption. *Crypto 2007*, *LNCS* 4622, pp. 517–534, 2007.
48. M. Prabhakaran, M. Rosulek. Homomorphic Encryption with CCA Security. *ICALP 2008*, *LNCS* 5126, pp. 667–678, 2008.
49. M. Prabhakaran, M. Rosulek. Towards Robust Computation on Encrypted Data. *Asiacrypt 2008*, *LNCS* 5350, pp. 216–233, 2008.
50. C. Rackoff, D. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *Crypto'91*, *LNCS* 576, pp. 433–444, 1991.
51. A. Sahai. Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen-Ciphertext Security. In *FOCS'99*, pp. 543–553, 1999.
52. H. Shacham. A Cramer-Shoup encryption scheme from the linear assumption and from progressively weaker linear variants. Cryptology ePrint Archive: Report 2007/074, 2007.
53. V. Shoup, R. Gennaro. Securing Threshold Cryptosystems against Chosen Ciphertext Attack. In *J. of Cryptology*, 15(2), pp. 75–96, 2002. Earlier version in *Eurocrypt'98*, *LNCS* 1403, pp. 1–16, 1998.
54. B. Waters. Efficient Identity-Based Encryption Without Random Oracles. In *Eurocrypt'05*, *LNCS* 3494, 2005.
55. H. Wee. Threshold and Revocation Cryptosystems via Extractable Hash Proofs. In *Eurocrypt'11*, *LNCS* 6632, pp. 589–609, 2011.

A Groth-Sahai Proofs

In the notations of this section, when vectors $\boldsymbol{A}$ and $\boldsymbol{B}$ are vectors of group elements, $\boldsymbol{A} \cdot \boldsymbol{B}$ denotes their entry-wise product.

Under the DLIN assumption in symmetric bilinear groups $(\mathbb{G}, \mathbb{G}_T)$, the Groth-Sahai (GS) proof systems [33] use a CRS consisting of three vectors $\boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{g}_3 \in \mathbb{G}^3$, where $\boldsymbol{g}_1 = (g_1, 1, g)$, $\boldsymbol{g}_2 = (1, g_2, g)$ for some $g_1, g_2 \in_R \mathbb{G}$. In this setting, a commitment to a group element $X \in \mathbb{G}$ is computed as $\boldsymbol{C} = (1,1,X) \cdot \boldsymbol{g_1}^r \cdot \boldsymbol{g_2}^s \cdot \boldsymbol{g_3}^t$ with $r, s, t \xleftarrow{R} \mathbb{Z}_p^*$. In order to obtain perfectly sound proofs, $\boldsymbol{g}_3$ is chosen as $\boldsymbol{g_3} = \boldsymbol{g_1}^{\xi_1} \cdot \boldsymbol{g_2}^{\xi_2}$, with $\xi_1, \xi_2 \xleftarrow{R} \mathbb{Z}_p^*$, so that the commitment $\boldsymbol{C} = (g_1^{r+\xi_1 t}, g_2^{s+\xi_2 t}, X \cdot g^{r+s+t(\xi_1+\xi_2)})$ is extractable as it is distributed as a Boneh-Boyen-Shacham (BBS) ciphertext [10] that can be decrypted using the discrete logarithms $\alpha_1 = \log_g(g_1)$, $\alpha_2 = \log_g(g_2)$. In order to switch to the witness indistinguishability setting, the vectors $\boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{g}_3$ must be linearly independent so as to span the entire space where $\boldsymbol{C}$ lives and make sure that $\boldsymbol{C}$ is a perfectly hiding commitment. Under the DLIN assumption, the two kinds of reference strings are computationally indistinguishable.

When it comes to commit to an exponent $x \in \mathbb{Z}_p$, the prover computes $\boldsymbol{C} = \boldsymbol{\varphi}^x \cdot \boldsymbol{g}_1^r \cdot \boldsymbol{g}_2^s$, with $r, s \xleftarrow{R} \mathbb{Z}_p^*$, using a CRS comprising vectors $\boldsymbol{\varphi}, \boldsymbol{g}_1, \boldsymbol{g}_2$. In the soundness setting $\boldsymbol{\varphi}, \boldsymbol{g}_1, \boldsymbol{g}_2$ are linearly independent vectors while, in the perfect WI setting, choosing $\boldsymbol{\varphi}$ in $\text{span}(\boldsymbol{g}_1, \boldsymbol{g}_2)$ yields a perfectly hiding commitment as $\boldsymbol{C}$ is always a BBS encryption of $1_{\mathbb{G}}$.

To prove that committed group elements or exponents satisfy certain relations, the Groth-Sahai methodology [33] requires one commitment per variable and one proof element per relation. Efficient NIWI proofs are available for multi-exponentiation equations of the form

$$\prod_{i=1}^{m} \mathcal{A}_i^{y_i} \cdot \prod_{j=1}^{n} \mathcal{X}_j^{b_j} \cdot \prod_{i=1}^{m} \cdot \prod_{j=1}^{n} \mathcal{X}_j^{y_i \gamma_{ij}} = T,$$

for variables $\mathcal{X}_1, \ldots, \mathcal{X}_n \in \mathbb{G}$, $y_1, \ldots, y_m \in \mathbb{Z}_p$ and constants $T, \mathcal{A}_1, \ldots, \mathcal{A}_m \in \mathbb{G}$, $b_1, \ldots, b_n \in \mathbb{Z}_p$ and $\gamma_{ij} \in \mathbb{Z}_p$, for $i \in \{1, \ldots, m\}, j \in \{1, \ldots, n\}$.

For linear equations (*i.e.*, where $\gamma_{ij} = 0$ for all i, j) depends on the form of the considered equation.

Namely, linear multi-exponentiation equations of the type $\prod_{j=1}^{n} \mathcal{X}_j^{b_j} = T$ (resp. $\prod_{i=1}^{m} \mathcal{A}_i^{y_i} = T$) demand 3 (resp. 2) group elements.

Multi-exponentiation equations admit NIZK proofs. On a simulated CRS, the representation $(\xi_1, \xi_2) \in \mathbb{Z}_p^2$ of $\boldsymbol{\varphi}$ as $\boldsymbol{\varphi} = \boldsymbol{g}_1^{\xi_1} \cdot \boldsymbol{g}_2^{\xi_2}$ can serve as a trapdoor that makes it possible to perfectly simulate proofs without knowing the witnesses.

Efficient NIWI proofs also exist for pairing-product relations, which are the form

$$\prod_{i=1}^{n} e(\mathcal{A}_i, \mathcal{X}_i) \cdot \prod_{i=1}^{n} \cdot \prod_{j=1}^{n} e(\mathcal{X}_i, \mathcal{X}_j)^{a_{ij}} = t_T,$$

for variables $\mathcal{X}_1, \ldots, \mathcal{X}_n \in \mathbb{G}$ and constants $t_T \in \mathbb{G}_T$, $\mathcal{A}_1, \ldots, \mathcal{A}_n \in \mathbb{G}$, $a_{ij} \in \mathbb{Z}_p$, for $i, j \in \{1, \ldots, n\}$. For each linear pairing product equation (where $a_{ij} = 0$ for all i, j), a proof fits within 3 group elements. Quadratic equations are somewhat more space-consuming and take 9 group elements each. At the cost of introducing extra variables, pairing product equations can also have NIZK proofs.

B Definitions for Linearly Homomorphic Structure-Preserving Signatures

Let $(\mathbb{G}, \mathbb{G}_T)$ be groups of prime order p such that a bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ can be efficiently computed.

A signature scheme is *structure-preserving* [3, 2] if messages, signatures and public keys all live in the group $\mathbb{G}$. In linearly homomorphic structure-preserving signatures, the message space $\mathcal{M}$ consists of pairs $\mathcal{M} := \mathcal{T} \times \mathbb{G}^n$, for some $n \in \mathbb{N}$, where $\mathcal{T}$ is a tag space. Depending on the application, one may want the tags to be group elements or not. In this paper, they can be arbitrary strings.

Definition 4. *A linearly homomorphic structure-preserving signature scheme over $(\mathbb{G}, \mathbb{G}_T)$ is a tuple of efficient algorithms $\Sigma = (\mathsf{Keygen}, \mathsf{Sign}, \mathsf{SignDerive}, \mathsf{Verify})$ for which the message space consists of $\mathcal{M} := \mathcal{T} \times \mathbb{G}^n$, for some integer $n \in \mathsf{poly}(\lambda)$ and some set $\mathcal{T}$, and with the following specifications.*

$\mathsf{Keygen}(\lambda, n)$: *is a randomized algorithm that takes in a security parameter $\lambda \in \mathbb{N}$ and an integer $n \in \mathsf{poly}(\lambda)$ denoting the dimension of vectors to be signed. It outputs a key pair $(\mathsf{pk}, \mathsf{sk})$, where pk includes the description of a tag space $\mathcal{T}$, where each tag serves as a file identifier.*

$\mathsf{Sign}(\mathsf{sk}, \tau, \boldsymbol{M})$: *is a possibly randomized algorithm that takes as input a private key sk, a file identifier $\tau \in \mathcal{T}$ and a vector $\boldsymbol{M} = (M_1, \ldots, M_n) \in \mathbb{G}^n$. It outputs a signature $\sigma \in \mathbb{G}^{n_s}$, for some $n_s \in \mathsf{poly}(\lambda)$.*

$\mathsf{SignDerive}(\mathsf{pk}, \tau, \{(\omega_i, \sigma^{(i)})\}_{i=1}^{\ell})$: *is a (possibly randomized) derivation algorithm. It inputs a public key pk, a file identifier τ as well as ℓ pairs $(\omega_i, \sigma^{(i)})$, each of which consists of a coefficient $\omega_i \in \mathbb{Z}_p$ and a signature $\sigma^{(i)} \in \mathbb{G}^{n_s}$. It outputs a signature $\sigma \in \mathbb{G}^{n_s}$ on the vector $\boldsymbol{M} = \prod_{i=1}^{\ell} \boldsymbol{M}_i{}^{\omega_i}$, where $\sigma^{(i)}$ is a signature on $\boldsymbol{M}_i$.*

$\mathsf{Verify}(\mathsf{pk}, \tau, \boldsymbol{M}, \boldsymbol{\sigma})$: *is a deterministic verification algorithm that takes as input a public key pk, a file identifier $\tau \in \mathcal{T}$, a signature σ and a vector $\boldsymbol{M} = (M_1, \ldots, M_n)$. It outputs 0 or 1 depending on whether σ is deemed valid or not.*

In a *one-time* linearly homomorphic SPS, the tag τ can be omitted in the specification as a given key pair $(\mathsf{pk}, \mathsf{sk})$ only allows signing one linear subspace.

As in all linearly homomorphic signatures, the security requirement is that the adversary be unable to create a valid triple $(\tau^\star, \boldsymbol{M}^\star, \sigma^\star)$ for a new file identifier $\tau^\star$ or, if $\tau^\star$ is recycled from one or more honestly generated signatures, for a vector $\boldsymbol{M}^\star$ outside the linear span of the vectors that have been legitimately signed for the tag $\tau^\star$.

An important property is that the $\mathsf{SignDerive}$ algorithm must operate on vectors that are all labeled with the same tag.

C Randomizable Linearly Homomorphic Structure-Preserving Signatures

This section recalls the randomizable linearly homomorphic structure-preserving signature of [43].

In the scheme, each signature basically consists of a Groth-Sahai NIWI proof of knowledge of a one-time signature (z, r, u) on the signed vector $(M_1, \ldots, M_n)$. This proof of knowledge is generated for a Groth-Sahai CRS which depends on the tag that identifies the subspace being signed.

In the following notations, for each $h \in \mathbb{G}$ and any vector $\boldsymbol{g} = (g_1, g_2, g_3) \in \mathbb{G}^3$, we denote by $E(h, \boldsymbol{g})$ the vector $(e(h, g_1), e(h, g_2), e(h, g_3)) \in \mathbb{G}_T^3$.

Keygen($\boldsymbol{\lambda}, \boldsymbol{n}$): given a security parameter λ and the dimension $n \in \mathbb{N}$ of the subspace to be signed, choose bilinear group $(\mathbb{G}, \mathbb{G}_T)$ of order $p > 2^\lambda$ with a generator $g \stackrel{R}{\leftarrow} \mathbb{G}$ as well as $g_z, g_r, h_z, h_u \stackrel{R}{\leftarrow} \mathbb{G}$ and do the following.

1. For $i = 1$ to n, pick $\chi_i, \gamma_i, \delta_i \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and compute $g_i = g_z{}^{\chi_i} g_r{}^{\gamma_i}$ and $h_i = h_z{}^{\chi_i} h_u{}^{\delta_i}$.
2. Generate $L+1$ Groth-Sahai common reference strings, where $L \in \mathsf{poly}(\lambda)$ is the length of each tag $\tau \in \mathcal{T} = \{0,1\}^L$. To this end, choose $f_1, f_2 \stackrel{R}{\leftarrow} \mathbb{G}$ and define vectors $\boldsymbol{f}_1 = (f_1, 1, g) \in \mathbb{G}^3$, $\boldsymbol{f}_2 = (1, f_2, g) \in \mathbb{G}^3$. Then, pick $\boldsymbol{f}_{3,i} \stackrel{R}{\leftarrow} \mathbb{G}^3$ for $i = 0$ to L.

The public key consists of

$$\mathsf{pk} = \Big(\ (\mathbb{G}, \mathbb{G}_T),\ g_z,\ g_r,\ h_z,\ h_u,\ \{g_i, h_i\}_{i=1}^n,\ \mathbf{f} = \big(\boldsymbol{f}_1, \boldsymbol{f}_2, \{\boldsymbol{f}_{3,i}\}_{i=0}^L\big)\ \Big)$$

while the private key is $\mathsf{sk} = \big(h_z{}^{\alpha_r}, \{\chi_i, \gamma_i, \delta_i\}_{i=1}^n\big)$.

Sign($\mathsf{sk}, \boldsymbol{\tau}, (\boldsymbol{M_1}, \ldots, \boldsymbol{M_n})$): to sign a vector $(M_1, \ldots, M_n) \in \mathbb{G}^n$ using $\mathsf{sk} = \big(h_z{}^{\alpha_r},\ \{\chi_i, \gamma_i, \delta_i\}_{i=1}^n\big)$ with the file identifier τ, conduct the following steps.

1. Choose $\theta \stackrel{R}{\leftarrow} \mathbb{Z}_p$ at random and compute $z = g_r{}^{\theta} \cdot \prod_{i=1}^n M_i^{-\chi_i}$, $r = g_z{}^{-\theta} \cdot \prod_{i=1}^n M_i^{-\gamma_i}$ and $u = h_z{}^{-\theta \cdot \alpha_r} \cdot \prod_{i=1}^n M_i^{-\delta_i}$.
2. Using the bits $\tau[1] \ldots \tau[L]$ of $\tau \in \{0,1\}^L$, define the vector $\boldsymbol{f}_\tau = \boldsymbol{f}_{3,0} \cdot \prod_{i=1}^L \boldsymbol{f}_{3,i}^{\tau[i]}$ so as to assemble a Groth-Sahai CRS $\mathbf{f}_\tau = (\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_\tau)$.
3. Using $\mathbf{f}_\tau = (\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_\tau)$, compute commitments $\boldsymbol{C}_z = (1_{\mathbb{G}}, 1_{\mathbb{G}}, z) \cdot \boldsymbol{f}_1^{\nu_{z,1}} \cdot \boldsymbol{f}_2^{\nu_{z,2}} \cdot \boldsymbol{f}_\tau^{\nu_{z,3}}$ and

   $$\boldsymbol{C}_r = (1_{\mathbb{G}}, 1_{\mathbb{G}}, r) \cdot \boldsymbol{f}_1^{\nu_{r,1}} \cdot \boldsymbol{f}_2^{\nu_{r,2}} \cdot \boldsymbol{f}_\tau^{\nu_{r,3}} \qquad \boldsymbol{C}_u = (1_{\mathbb{G}}, 1_{\mathbb{G}}, u) \cdot \boldsymbol{f}_1^{\nu_{u,1}} \cdot \boldsymbol{f}_2^{\nu_{u,2}} \cdot \boldsymbol{f}_\tau^{\nu_{u,3}}$$

   to the derived z, r and u, respectively. Generate NIWI proofs $\boldsymbol{\pi}_1 = (\pi_{1,1}, \pi_{1,2}, \pi_{1,3}) \in \mathbb{G}^3$ and $\boldsymbol{\pi}_2 = (\pi_{2,1}, \pi_{2,2}, \pi_{2,3}) \in \mathbb{G}^3$ that (z, r, u) satisfy the verification equations

   $$1_{\mathbb{G}_T} = e(g_z, z) \cdot e(g_r, r) \cdot \prod_{i=1}^n e(g_i, M_i) \qquad 1_{\mathbb{G}_T} = e(h_z, z) \cdot e(h_u, u) \cdot \prod_{i=1}^n e(h_i, M_i).$$

   These proofs are obtained as

   $$\begin{aligned} \boldsymbol{\pi}_1 &= (\pi_{1,1}, \pi_{1,2}, \pi_{1,3}) = \big(g_z{}^{-\nu_{z,1}} \cdot g_r{}^{-\nu_{r,1}},\ g_z{}^{-\nu_{z,2}} \cdot g_r{}^{-\nu_{r,2}},\ g_z{}^{-\nu_{z,3}} \cdot g_r{}^{-\nu_{r,3}}\big) \\ \boldsymbol{\pi}_2 &= (\pi_{2,1}, \pi_{2,2}, \pi_{2,3}) = \big(h_z{}^{-\nu_{z,1}} \cdot h_u{}^{-\nu_{u,1}},\ h_z{}^{-\nu_{z,2}} \cdot h_u{}^{-\nu_{u,2}},\ h_z{}^{-\nu_{z,3}} \cdot h_u{}^{-\nu_{u,3}}\big). \end{aligned}$$

The signature consists of $\sigma = (\boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2) \in \mathbb{G}^{15}$.

SignDerive($\mathsf{pk}, \boldsymbol{\tau}, \{(\boldsymbol{\omega_i}, \boldsymbol{\sigma^{(i)}})\}_{\boldsymbol{i=1}}^{\boldsymbol{\ell}}$): given pk, a file identifier τ and ℓ tuples $(\omega_i, \sigma^{(i)})$, parse each signature $\sigma^{(i)}$ as a tuple of the form $\sigma^{(i)} = (\boldsymbol{C}_{z,i}, \boldsymbol{C}_{r,i}, \boldsymbol{C}_{u,i}, \boldsymbol{\pi}_{1,i}, \boldsymbol{\pi}_{2,i}) \in \mathbb{G}^{15}$ for $i = 1$ to ℓ.

1. Compute $\boldsymbol{C}_z = \prod_{i=1}^\ell \boldsymbol{C}_{z,i}^{\omega_i}$, $\boldsymbol{C}_r = \prod_{i=1}^\ell \boldsymbol{C}_{r,i}^{\omega_i}$, $\boldsymbol{C}_u = \prod_{i=1}^\ell \boldsymbol{C}_{u,i}^{\omega_i}$, $\boldsymbol{\pi}_1 = \prod_{i=1}^\ell \boldsymbol{\pi}_{1,i}^{\omega_i}$ as well as $\boldsymbol{\pi}_2 = \prod_{i=1}^\ell \boldsymbol{\pi}_{2,i}^{\omega_i}$.
2. Re-randomize the above commitments and proofs and return $\sigma = (\boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$.

Verify(pk, $\boldsymbol{\sigma}, \boldsymbol{\tau}, (\boldsymbol{M_1}, \ldots, \boldsymbol{M_n})$): given $(\tau, (M_1, \ldots, M_n))$ and a purported signature σ, parse σ as $(\boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$. Return 1 iff $(M_1, \ldots, M_n) \neq (1_{\mathbb{G}}, \ldots, 1_{\mathbb{G}})$ and the proofs $(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$ satisfy

$$\begin{aligned}
\prod_{i=1}^{n} E\big(g_i, (1_{\mathbb{G}}, 1_{\mathbb{G}}, M_i)\big)^{-1} &= E\big(g_z, \boldsymbol{C}_z\big) \cdot E\big(g_r, \boldsymbol{C}_r\big) \cdot E(\pi_{1,1}, \boldsymbol{f}_1) \cdot E(\pi_{1,2}, \boldsymbol{f}_2) \cdot E(\pi_{1,3}, \boldsymbol{f}_\tau) \qquad (4)\\
\prod_{i=1}^{n} E\big(h_i, (1_{\mathbb{G}}, 1_{\mathbb{G}}, M_i)\big)^{-1} &= E\big(h_z, \boldsymbol{C}_z\big) \cdot E\big(h_u, \boldsymbol{C}_u\big) \cdot E(\pi_{2,1}, \boldsymbol{f}_1) \cdot E(\pi_{2,2}, \boldsymbol{f}_2) \cdot E(\pi_{2,3}, \boldsymbol{f}_\tau).
\end{aligned}$$

We remark that the scheme can be simplified by setting $\theta = 0$ in all algorithms: since all non-interactive proofs are generated for a perfectly NIWI Groth-Sahai CRS, this modification does not affect the distribution of signatures whatsoever. In Sections 3 and 5, we use this simplified version of the scheme.

The scheme is only known [43] to be secure in a relaxed model where the adversary is only deemed successful if it additionally provides evidence that its output vector is indeed independent of those for which it obtained signatures with respect to the target tag $\tau^\star$. In our applications, this restriction will not be a problem at all since, in all security proofs, the reduction will always be able to detect when the adversary has won without requiring explicit evidence for it.

D Extensions Based on the k-Linear Assumption

To instantiate our proof systems using the k-linear assumption with $k > 2$, we first need to extend the one-time linearly homomorphic structure-preserving signature of [43]. To this end, we need to define the following assumption which is implied by the k-linear assumption in the same way as SDP is implied by DLIN.

Definition 5. *The* **Simultaneous k-wise Pairing** *(k-SDP) problem is, given a random tuple*

$$(g_{1,z}, \ldots, g_{k,z}, g_{1,r}, \ldots, g_{k,r}) \in_R \mathbb{G}^{2k},$$

to find a non-trivial vector $(z, r_1, \ldots, r_k) \in \mathbb{G}^{k+1}$ such that

$$e(g_{j,z}, z) \cdot e(g_{j,r}, r_j) = 1_{\mathbb{G}_T} \qquad j \in \{1, \ldots, k\} \qquad (5)$$

and $z \neq 1_{\mathbb{G}}$.

Given a k-linear instance $(g_{1,r}, \ldots, g_{k,r}, g_{1,r}^{a_1}, \ldots, g_{k,r}^{a_k}, \eta) \in \mathbb{G}^{2k+1}$, for any non-trivial tuple $(z, r_1, \ldots, r_k)$ satisfying $e(g_{j,r}^{a_j}, z) \cdot e(g_{j,r}, r_j) = 1_{\mathbb{G}_T}$ for each $j \in \{1, \ldots, k\}$, we have

$$\eta = g^{\sum_{j=1}^{k} a_j} \qquad \Leftrightarrow \qquad e(g, \prod_{j=1}^{k} r_j) \cdot e(z, \eta) = 1_{\mathbb{G}_T}.$$

Hence, any algorithm solving k-SDP with non-negligible probability implies a k-linear distinguisher.

Under the k-SDP assumption, the one-time linearly homomorphic structure-preserving signature of [43] can be extended as follows.

Keygen($\boldsymbol{\lambda}, \boldsymbol{n}$): given a security parameter λ and the dimension $n \in \mathbb{N}$ of vectors to be signed, choose bilinear group $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$. For $j = 1$ to k, choose generators $g_{j,z}, g_{j,r} \stackrel{R}{\leftarrow} \mathbb{G}$. Then, for each $i = 1$ to n, $j = 1$ to k, pick $\chi_i \stackrel{R}{\leftarrow} \mathbb{Z}_p$, $\gamma_{j,i} \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and compute $g_{j,i} = g_{j,z}{}^{\chi_i} g_{j,r}{}^{\gamma_{j,i}}$. The private key is $\mathsf{sk} = \big(\{\chi_i, \{\gamma_{j,i}\}_{j=1}^{k}\}_{i=1}^{n}\big)$ while the public key is

$$\mathsf{pk} = \Big(\{g_{j,z}, g_{j,r}, \{g_{j,i}\}_{i=1}^{n}\}_{j=1}^{k}\Big).$$

Sign(sk, $(\boldsymbol{M_1, \ldots, M_n})$): to sign $(M_1, \ldots, M_n) \in \mathbb{G}^n$ using $\mathsf{sk} = \big(\{\chi_i, \{\gamma_{j,i}\}_{j=1}^{k}\}_{i=1}^{n}\big)$, compute and output $\sigma = (z, r_1, \ldots, r_k) \in \mathbb{G}^{k+1}$, where

$$z = \prod_{i=1}^{n} M_i^{-\chi_i},$$
$$r_j = \prod_{i=1}^{n} M_i^{-\gamma_{j,i}} \qquad\qquad j \in \{1, \ldots, k\}.$$

SignDerive(pk, $\{(\boldsymbol{\omega_i, \sigma^{(i)}})\}_{\boldsymbol{i=1}}^{\boldsymbol{\ell}}$): given a public key pk and ℓ tuples $(\omega_i, \sigma^{(i)})$, where $\omega_i \in \mathbb{Z}_p$ for each i, parse $\sigma^{(i)}$ as $\sigma^{(i)} = \big(z_i, r_{i,1}, \ldots, r_{i,k}\big) \in \mathbb{G}^{k+1}$ for $i = 1$ to ℓ. Then, compute and return $\sigma = (z, r_1, \ldots, r_k)$, where $z = \prod_{i=1}^{\ell} z_i^{\omega_i}$, $r_j = \prod_{i=1}^{\ell} r_{i,j}^{\omega_i}$ for $j = 1$ to k.

Verify(pk, $\boldsymbol{\sigma}$, $(\boldsymbol{M_1, \ldots, M_n})$): given $\sigma = (z, r_1, \ldots, r_k) \in \mathbb{G}^{k+1}$ and $(M_1, \ldots, M_n)$, return 1 if and only if $(M_1, \ldots, M_n) \neq (1_{\mathbb{G}}, \ldots, 1_{\mathbb{G}})$ and, for each $j \in \{1, \ldots, k\}$, the following equality holds:

$$1_{\mathbb{G}_T} = e(g_{j,z}, z) \cdot e(g_{j,r}, r_j) \cdot \prod_{i=1}^{n} e(g_{j,i}, M_i). \tag{6}$$

In order to adapt the unbounded simulation-sound proof system of Section 3, we need to commit to the components of $(z, r_1, \ldots, r_k)$ and NIWI arguments showing that committed group elements satisfy the pairing product equations (6). Under the k-linear assumption, committing to a group element requires $k+1$ group elements (see, *e.g.*, [15] for details) whereas each equation of the form (6) costs $k+1$ elements to prove. Overall, we thus need $(k+1)(2k+1)$ group elements and a one-time verification key pair (SVK, σ).

In the relatively-sound QA-NIZK proof of Section 4, the proof element π_0 remains unchanged and we simply need to replace the triple (z, r, u) by a one-time linearly homomorphic signature $(z, r_1, \ldots, r_k)$. Hence, we only need $k+2$ group elements.

E Comparisons

This section compares the various NIZK proofs of linear subspace membership based on the DLIN assumption. Comparisons are given in terms of CRS size, proof size, the number of pairing evaluations for the verifier and the need for a computational assumption to prove the soundness property.

In the table, we consider our basic proof system (without any form of simulation-soundness, where each proof is a one-time linearly homomorphic signature (z, r, u)), its unbounded simulation-sound variant and the relatively simulation-sound variant of Section 4. We compare these with the original Groth-Sahai proofs, their most efficient unbounded simulation-sound extensions due to Camenisch *et al.* [15] and the Jutla-Roy techniques [38] with and without relative soundness.

Table 1. Comparison between proof systems for linear subspaces

Proof systems	CRS size◇ *	Proof length◇	# of pairings† at verification	Soundness property
Groth-Sahai [33]	6	$3t + 2n$	$3n(t+3)$	perfect
Jutla-Roy [38]	$4t(n-t)+3$	$2(n-t)$	2(n-t)(t+2)	computational
Jutla-Roy RSS [38] + [37]	$4t(n+1-t)+3$	$2(n+1-t)+1$	$2(n+1-t)(t+2)$	computational
Groth-Sahai USS [15]	18	$6t + 2n + 52$‡	$O(tn)$	computational
Our basic QA-NIZK proofs	$2n + 3t + 4$	3	$2n + 4$	computational
Our RSS QA-NIZK proofs	$4n + 8t + 6$	4	$2n + 6$	computational
Our USS QA-NIZK proofs	$2n + 3t + 3L + 10$	20‡	$2n + 30$	computational

n: number of equations; t: number of variables; L: length of a hashed one-time verification key

◇ These sizes are measured in terms of number of group elements.

∗ The description $\rho \in \mathbb{G}^{t \times n}$ of the language is not counted as being part of the CRS here.

† The table does not consider optimizations using randomized batch verification techniques here.

‡ We consider instantiations using Groth's one-time signature [32], where verification keys and signatures consist of 3 group elements and two elements of $\mathbb{Z}_p$, respectively.

As can be observed in the table, our constructions all yield constant-size arguments. Moreover, the number of pairing evaluations is always independent of the number of variables t, which substantially fastens the verification process when $t \approx n/2$.

We also note that randomized batch verification techniques can be used to drastically reduce the number of pairing computations. In our USS system, for example, the number of pairings drops to $n + 18$ if the two verification equations are processed together and further optimizations are possible.

Our common reference strings always fit within $O(t + n)$ group elements (with another $O(L)$ elements in the USS variant) and thus provide significant savings w.r.t. [38] when $t \approx n/2$.

F Proof of Theorem 1

Proof. The quasi-adaptive completeness property follows directly from the correctness of the randomizable linearly homomorphic signature of Section 2.5. We thus focus on the quasi-adaptive zero-knowledge and quasi-adaptive unbounded simulation-soundness properties.

Quasi-Adaptive Zero-Knowledge. To prove this property we consider a sequence of two games which begins with a game where the adversary has access to a real prover P on a real CRS ψ. In the second game, the adversary will be faced with a simulator $(\mathsf{S}_1, \mathsf{S}_2)$.

Game_1**:** is a game where the adversary $\mathcal{A}$ is given the description of the language $\mathcal{L}_\rho$ and is granted access to a real CRS ψ and an actual prover $\mathsf{P}(\psi, ., .)$ which takes as input a vector $\boldsymbol{v}$ along with a witness $\boldsymbol{x} \in \mathbb{Z}_p^t$ such that $\boldsymbol{v} = g^{\boldsymbol{x} \cdot \mathbf{A}}$. At each invocation, the oracle outputs a genuine proof π by running the legal P algorithm. The adversary is allowed to query $\mathsf{P}(\psi, ., .)$ a polynomial number of times and eventually outputs a bit $\beta \in \{0, 1\}$. We denote by S_1 the event that $\beta = 1$.

Game_2**:** This game is identical to Game_2 with the difference that, when the $\mathsf{P}(\psi, ., .)$ oracle is queried on a pair $(\boldsymbol{v}, \boldsymbol{x})$, it does not use the witness $\boldsymbol{x} \in \mathbb{Z}_p^t$ anymore at step 1 of the proving algorithm. Instead, it uses the the private key $\mathsf{sk}_{rand} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^n$ to compute a one-time signature $(z, r, u) = \big(\prod_{j=1}^n v_j^{-\chi_j}, \prod_{j=1}^n v_j^{-\gamma_j}, \prod_{j=1}^n v_j^{-\delta_j}\big)$ on the vector $\boldsymbol{v} = (v_1, \ldots, v_n) \in \mathbb{G}^n$. The remaining parts of π are generated as in the real $\mathsf{P}(\psi, ., .)$ oracle in steps 2 and 3 of the proof generation algorithm. Although, the witness $\boldsymbol{x} \in \mathbb{Z}_p^t$ is never used, it is easy to see that (z, r, u) has exactly the same distribution as in Game_2 if $\boldsymbol{v} \in \mathcal{L}_\rho$ (*i.e.*, as long as $\boldsymbol{v} = g^{\boldsymbol{x} \cdot \mathbf{A}}$ for some $\boldsymbol{x} \in \mathbb{Z}_p^t$). We thus have $\Pr[S_2] = \Pr[S_1]$.

We define the simulator $(\mathsf{S}_1, \mathsf{S}_2)$ by having S_1 generate the CRS ψ as in Game_1 (so that ψ has the same distribution as the real CRS) and letting S_2 generate proofs without using the witnesses as in Game_2. It easily comes that the system is perfectly quasi-adaptive zero-knowledge as, for all language members $\boldsymbol{v} \in \mathbb{G}^n$, simulated proofs are distributed as real proofs.

Quasi-Adaptive Unbounded Simulation-Soundness. To prove this property, we proceed again with a sequence of games.

Game_1**:** is the real game where the adversary $\mathcal{A}$ is given the description of the language $\mathcal{L}_\rho$ and is granted access to a simulated CRS ψ and a simulated prover $\mathsf{S}_2(\psi, \tau_{sim}, ., .)$ which takes as input a vector-label pair $(\boldsymbol{v}, \mathsf{lbl})$ and returns a simulated proof π that $\boldsymbol{v} \in \mathcal{L}_\rho$. To generate $\rho \in \mathbb{G}^{t \times n}$ according to the distribution D_Γ, the challenger generates a matrix $\mathbf{A} \in \mathbb{Z}_p^{t \times n}$ with the appropriate distribution (recall that D_Γ is efficiently samplable by hypothesis) and computes $\rho = g^{\mathbf{A}}$. In addition, the challenger $\mathcal{B}$ computes a basis $\mathbf{W} \in \mathbb{Z}_p^{n \times (n-t)}$ of the right kernel of $\mathbf{A}$ and retains it for later use. The adversary is allowed to query the simulated prover $\mathsf{S}_2(\psi, \tau_{sim}, ., .)$ on polynomially many occasions. The game ends with the adversary $\mathcal{A}$ outputting an element $\boldsymbol{v}^\star$, a proof $\pi^\star$ and a label $\mathsf{lbl}^\star$. The adversary is deemed successful if $\boldsymbol{v}^\star \notin \mathcal{L}_\rho$ (*i.e.*, $\boldsymbol{v}^\star$ is not in the row space of

$\rho \in \mathbb{G}^{t\times n}$) but $(\pi^\star, \mathsf{lbl}^\star)$ is a valid proof. We denote by S_1 the latter event, which $\mathcal{B}$ can recognize by testing if $\boldsymbol{v} = (v_1, \ldots, v_n) \in \mathbb{G}^n$ satisfies $\prod_{j=1}^n v_j^{w_{ji}} = 1_{\mathbb{G}}$ for each column $\boldsymbol{w}_i^\top = (w_{1i}, \ldots, w_{ni})^\top$ of $\mathbf{W}$. Indeed, the vectors $\mathbf{y} \in \mathbb{Z}_p^n$ in the row space of $\mathbf{A}$ are exactly those for which $\mathbf{y} \cdot \mathbf{W} = \mathbf{0}$.

Game_2**:** This game is identical to Game_1 but the challenger $\mathcal{B}$ aborts if the adversary $\mathcal{A}$ outputs a fake proof $\pi^\star$ that recycles one of the one-time verification keys appearing in outputs of the $\mathsf{S}_2(\psi, \tau_{sim}, ., .)$ oracle. Clearly, Game_2 and Game_1 are identical until the latter event occurs and this event contradicts the strong unforgeability of Σ: if q denotes the number of queries to $\mathsf{S}_2(\psi, \tau_{sim}, ., .)$, a standard argument shows that $|\Pr[S_2] - \Pr[S_1]| \leq q \cdot \mathbf{Adv}^{\text{suf-ots}}(\mathcal{B})$.

Game_3**:** This game is identical to Game_2 but we modify the generation of pk_{rand} when the public key is set up. Namely, the vectors $(\boldsymbol{f}_1, \boldsymbol{f}_2, \{\boldsymbol{f}_{3,i}\}_{i=0}^L)$ are chosen by setting $\boldsymbol{f}_1 = (f_1, 1_{\mathbb{G}}, g)$ and $\boldsymbol{f}_2 = (1_{\mathbb{G}}, f_2, g)$ where $f_1, f_2 \xleftarrow{R} \mathbb{G}$ are chosen at random. As for $\{\boldsymbol{f}_{3,i}\}_{i=0}^L$, they are obtained as

$$\begin{aligned} \boldsymbol{f}_{3,0} &= \boldsymbol{f}_1^{\xi_{0,1}} \cdot \boldsymbol{f}_2^{\xi_{0,2}} \cdot (1,1,g)^{\xi_{0,3}} \cdot (1,1,g)^{\mu\cdot\zeta-\rho_0} && (7)\\ \boldsymbol{f}_{3,i} &= \boldsymbol{f}_1^{\xi_{i,1}} \cdot \boldsymbol{f}_2^{\xi_{i,2}} \cdot (1,1,g)^{\xi_{i,3}} \cdot (1,1,g)^{-\rho_i}, && i \in \{1,\ldots,L\} \end{aligned}$$

with $\mu \xleftarrow{R} \{0,\ldots,L\}$, $\xi_{0,1}, \xi_{1,1}, \ldots, \xi_{L,1} \xleftarrow{R} \mathbb{Z}_p$, $\xi_{0,2}, \xi_{1,2}, \ldots, \xi_{L,2} \xleftarrow{R} \mathbb{Z}_p$, $\xi_{0,3}, \xi_{1,3}, \ldots, \xi_{L,3} \xleftarrow{R} \mathbb{Z}_p$ and $\rho_0, \rho_1, \ldots, \rho_L \xleftarrow{R} \{0, \ldots, \zeta - 1\}$, with $\zeta = 2(q+1)$ and where q is the number of queries to the $\mathsf{S}_2(\psi, \tau_{sim}, .)$ oracle. Note that this change is only conceptual since the distribution of $\{\boldsymbol{f}_{3,i}\}_{i=0}^L$ has not changed since Game_2. We thus have $\Pr[S_3] = \Pr[S_2]$.

Game_4**:** This game is like Game_3 but we consider an event Good which causes the challenger $\mathcal{B}$ to abort if it does *not* occur. Let $\mathsf{SVK}_1, \ldots, \mathsf{SVK}_q$ be the distinct one-time verification keys appearing in outputs of the S_2 oracle throughout the game. Let also $\mathsf{SVK}^\star$ be the verification key involved in the fake proof $\pi^\star$ produced by $\mathcal{A}$. We know that $\mathsf{SVK}^\star \not\in \{\mathsf{SVK}_1, \ldots, \mathsf{SVK}_q\}$ unless the failure event introduced in Game_2 occurs. For each verification key $\mathsf{SVK} \in \{0,1\}^L$, we consider the function $J(\mathsf{SVK}) = \mu \cdot \zeta - \rho_0 - \sum_{i=1}^L \rho_i \cdot \mathsf{SVK}[i]$, where $\{\rho_i\}_{i=0}^L$ are the values internally defined by the simulator in Game_3. We also define Good to be the event that

$$J(\mathsf{SVK}^\star) = 0 \qquad \wedge \qquad \bigwedge_{j\in\{1,\ldots,q\}} J(\mathsf{SVK}_j) \neq 0. \tag{8}$$

We remark that the random exponents $\rho_0, \rho_1, \ldots, \rho_L$ are chosen independently of $\mathcal{A}$'s view: this means that the simulator could equivalently define $\{\boldsymbol{f}_{3,i}\}_{i=0}^L$ first and only choose $\{\rho_i\}_{i=0}^L$ – together with values $\{\xi_{3,i}\}_{i=0}^L$ explaining the $\{\boldsymbol{f}_{3,i}\}_{i=0}^L$ – at the very end of the game, when $\mathsf{SVK}^\star, \mathsf{SVK}_1, \ldots, \mathsf{SVK}_q, \mathsf{SVK}$ have been defined. The same analysis as [54] (using the simplifications of Bellare and Ristenpart [6, Theorem 3.1]) shows that $\Pr[S_4 \wedge \mathsf{Good}] \geq \Pr[S_3]^2/(27\cdot(q+1)\cdot(L+1))$.

This follows from the fact that, for any set of queries, a lower bound on the probability of event Good is $1/(2q(L+1))$. Indeed, from the known results [54, 35] on the programmability of Waters' hash function, we know that the probability, taken over the choice of $(\mu, \rho_0, \ldots, \rho_L)$, to meet the conditions (8) is at least $1/(2q(L+1))$.

Game_5**:** We modify again the way to compute pk_{rand} in the generation of the public key. Namely, the vectors $\boldsymbol{f_1} = (f_1, 1_{\mathbb{G}}, g)$, $\boldsymbol{f_2} = (1_{\mathbb{G}}, f_2, g)$ are chosen as before. However, instead of generating $\{\boldsymbol{f}_{3,i}\}_{i=0}^L$ as in Game_4, we set them as

$$\begin{aligned} \boldsymbol{f}_{3,0} &= \boldsymbol{f}_1^{\xi_{0,1}} \cdot \boldsymbol{f}_2^{\xi_{0,2}} \cdot (1,1,g)^{\mu\cdot\zeta-\rho_0} && (9)\\ \boldsymbol{f}_{3,i} &= \boldsymbol{f}_1^{\xi_{i,1}} \cdot \boldsymbol{f}_2^{\xi_{i,2}} \cdot (1,1,g)^{-\rho_i}, && i \in \{1,\ldots,L\} \end{aligned}$$

which amounts to setting $\xi_{0,3} = \xi_{1,3} = \ldots = \xi_{L,3} = 0$. Clearly, $\{\boldsymbol{f}_{3,i}\}_{i=0}^L$ are no longer uniform in the span of $\big(\boldsymbol{f}_1, \boldsymbol{f}_2, (1,1,g)\big)$. Still, this change should not be noticeable to $\mathcal{A}$ if the DLIN assumption holds in $\mathbb{G}$. Concretely, if the adversary wins (recall that the challenger can still detect this event using the matrix $\mathbf{W} \in \mathbb{Z}_p^{n\times(n-t)}$ as explained in Game_1) with substantially different probabilities in Game_5 and Game_4, we can construct a DLIN distinguisher $\mathcal{B}^{\mathrm{DLIN}}$ in the group $\mathbb{G}$. This distinguisher uses the random self-reducibility of DLIN to construct many independent-looking instances from the same distribution out of a given instance. The distinguisher then runs $\mathcal{A}$ on input of the CRS that was generated using the DLIN instances and it eventually outputs 1 if the adversary wins. We can thus write $|\Pr[S_5 \wedge \mathsf{Good}] - \Pr[S_4 \wedge \mathsf{Good}]| \leq \mathbf{Adv}_{\mathcal{B}^{\mathrm{DLIN}}}(\lambda)$.

In Game_5, we show that a successful adversary implies an algorithm $\mathcal{B}$ solving a given SDP instance (g_z, g_r, h_z, h_u) with non-negligible probability, which *a fortiori* breaks the DLIN assumption in $\mathbb{G}$.

By hypothesis, we know that $\mathcal{A}$ manages to create a proof $\pi^\star = (\mathsf{SVK}^\star, \boldsymbol{C}_z^\star, \boldsymbol{C}_r^\star, \boldsymbol{C}_u^\star, \boldsymbol{\pi}_1^\star, \boldsymbol{\pi}_2^\star, \sigma^\star)$ for a vector $\boldsymbol{v}^\star = (v_1^\star, \ldots, v_n^\star)$ outside the row space of ρ but $(\boldsymbol{C}_z^\star, \boldsymbol{C}_r^\star, \boldsymbol{C}_u^\star, \boldsymbol{\pi}_1^\star, \boldsymbol{\pi}_2^\star) \in \mathbb{G}^{15}$ and $\sigma^\star$ satisfy the verification equations. At this point, if the event Good introduced in Game_4 occurs, we must have $J(\mathsf{SVK}^\star) = 0$, which implies that $\boldsymbol{f}_{\mathsf{SVK}^\star} = \boldsymbol{f}_{3,0} \cdot \prod_{i=1}^{L+1} \boldsymbol{f}_{3,i}^{\mathsf{SVK}^\star[i]}$ lies in $\mathrm{span}(\boldsymbol{f}_1, \boldsymbol{f}_2)$. Consequently, $\boldsymbol{C}_z^\star$, $\boldsymbol{C}_r^\star$ and $\boldsymbol{C}_u^\star$ are necessarily perfectly binding and extractable commitments. Using $(\log_g(f_1), \log_g(f_2))$, algorithm $\mathcal{B}$ can thus extract the committed group elements $(z^\star, r^\star, u^\star) \in \mathbb{G}^3$ by BBS-decrypting the ciphertexts $(\boldsymbol{C}_z^\star, \boldsymbol{C}_r^\star, \boldsymbol{C}_u^\star)$. Since $(\boldsymbol{\pi}_1^\star, \boldsymbol{\pi}_2^\star)$ are perfectly sound Groth-Sahai proofs, the extracted elements $(z^\star, r^\star, u^\star)$ necessarily satisfy

$$1_{\mathbb{G}_T} = e(g_z, z^\star) \cdot e(g_r, r^\star) \cdot \prod_{i=1}^{n} e(g_i, v_i^\star) = e(h_z, z^\star) \cdot e(h_u, u^\star) \cdot \prod_{i=1}^{n} e(h_i, v_i^\star). \tag{10}$$

Having extracted $(z^\star, r^\star, u^\star)$, $\mathcal{B}$ also computes

$$z^\dagger = \prod_{i=1}^{n} v_i^{\star-\chi_i} \qquad r^\dagger = \prod_{i=1}^{n} v_i^{\star-\gamma_i} \qquad u^\dagger = \prod_{i=1}^{n} v_i^{\star-\delta_i}, \tag{11}$$

so that $(z^\dagger, r^\dagger, u^\dagger)$ also satisfies (10). Since $(z^\dagger, r^\dagger, u^\dagger)$ and $(z^\star, r^\star, u^\star)$ both satisfy (10), the triple

$$(z^\ddagger, r^\ddagger, u^\ddagger) = \Big(\frac{z^\star}{z^\dagger}, \frac{r^\star}{r^\dagger}, \frac{u^\star}{u^\dagger}\Big)$$

necessarily satisfies $e(g_z, z^\ddagger) \cdot e(g_r, r^\ddagger) = e(h_z, z^\ddagger) \cdot e(h_u, u^\ddagger) = 1_{\mathbb{G}_T}$. To conclude the proof, we argue that $z^\ddagger \neq 1_{\mathbb{G}}$ with overwhelming probability.

To do this, we observe that, if the event Good defined in Game_4 actually comes about, then $\mathcal{B}$ never leaks any more information about $(\chi_1, \ldots, \chi_n)$ than $\mathcal{A}$ can infer by just observing $\{(z_j, r_j, u_j)\}_{j=1}^{t}$ in the public key. Indeed, in this case we have $J(\mathsf{SVK}^\star) = 0$ and $J(\mathsf{SVK}_j) \neq 0$ for each $j \in \{1, \ldots, q\}$. This means that, in the simulated proofs returned by $\mathsf{S}_2(\psi, \tau_{sim}, ., .)$, the proofs $(\boldsymbol{\pi_1}, \boldsymbol{\pi_2})$ are perfectly witness indistinguishable as they are generated for a perfectly hiding Groth-Sahai CRS. For these simulated proofs, the built-in homomorphic signatures $(\boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi_2})$ leak nothing about the specific vector $(\chi_1, \ldots, \chi_n)$ used by $\mathcal{B}$. As a consequence, the same arguments as in [43, Theorem 1] show that $z^\dagger \neq z^\star$ with probability $1 - 1/p$. Specifically, in the CRS, $\{(g_i, h_i)\}_{i=1}^{n}$ and $\{(z_i, r_i, u_i)\}_{i=1}^{t}$ provide the adversary with a system of $2n + t < 3n$ equations in $3n$ unknowns $\{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^{n}$, which leaves $z^\dagger$ completely undetermined as long as $\boldsymbol{v}^\star$ is linearly independent of the rows of $\big(G_{i,j}\big)_{i,j}$. We thus find

$$\Pr[S_5 \wedge \mathsf{Good}] = \mathbf{Adv}_{\mathcal{B}}^{\mathrm{SDP}}(\lambda) \cdot \Big(1 - \frac{1}{p}\Big)^{-1}.$$

In turn, in Game_5, $\mathcal{B}$ implies a PPT distinguisher $\mathcal{B}^{\mathrm{DLIN}'}$ for the DLIN assumption such that we have the inequality $\Pr[S_5 \wedge \mathsf{Good}] < \frac{1}{2} \cdot \mathbf{Adv}_{\mathcal{B}^{\mathrm{DLIN}'}}(\lambda) \cdot \big(1 - \frac{1}{p}\big)^{-1}$. If $\mathbf{Adv}^{\mathrm{DLIN}}(\lambda)$ denotes the maximal advantage of any PPT distinguisher against the DLIN assumption in $\mathbb{G}$, the probability of event

$S_4 \wedge \mathsf{Good}$ can be bounded as $\Pr[S_4 \wedge \mathsf{Good}] \leq \frac{3}{2} \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\lambda) \cdot \big(1 - \frac{1}{p}\big)^{-1}$ in Game_5 in Game_4. This in turn yields

$$\Pr[S_3] \leq 7 \cdot \sqrt{q \cdot (L+1) \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\lambda) \cdot \big(1 - \frac{1}{p}\big)^{-1}},$$

so that $\mathcal{A}$'s advantage in breaking the unbounded simulation-soundness of the system is at most

$$\mathbf{Adv}^{\mathrm{uss}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{suf-ots}}(\lambda) + 7 \cdot \sqrt{q \cdot (L+1) \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\lambda) \cdot \big(1 - \frac{1}{p}\big)^{-1}}. \tag{12}$$

□

G Proof of Theorem 2

Proof. The completeness property is straightforward to verify. To establish the result, we separately prove the relative quasi-adaptive zero-knowledge and relative quasi-adaptive simulation-soundness properties.

Quasi-Adaptive Relative Zero-Knowledge. We consider a sequence of two games which begins with a game where the adversary has oracle access to the actual prover P and a public verifier on a real CRS ψ. In the last game, the adversary will be interacting with a simulator $(\mathsf{S}_1, \mathsf{S}_2)$ and the private verifier.

Game_1**:** is a game where the adversary $\mathcal{A}$ is given the description ρ of the language $\mathcal{L}_\rho$ and a real CRS $\psi = (\mathbf{CRS}_1, \mathbf{CRS}_2)$. In addition, the adversary has access to a public verification oracle $\mathsf{V}(\psi, ., .)$, even though it can run the verification algorithm by itself. This will be useful to show that the private verifier always agrees with the public one when it interacts with a PPT adversary. At some point, the adversary chooses a pair $(\boldsymbol{v}, \mathsf{lbl})$ along with a witness $\boldsymbol{x} \in \mathbb{Z}_p^t$ such that $\boldsymbol{v} = g^{\boldsymbol{x} \cdot \mathbf{A}}$. The challenger replies by returning an actual proof π produced by running $\mathsf{P}(\psi, \boldsymbol{v}, \boldsymbol{x}, \mathsf{lbl})$. When $\mathcal{A}$ terminates, it outputs a bit $\beta \in \{0, 1\}$. We denote by S_1 the event that $\beta = 1$.

Game_2**:** is like Game_1 but the adversary's public verification oracle $\mathsf{V}(\psi, ., .)$ is replaced by the private verification oracle $\mathsf{W}(\psi, \tau_v, ., .)$. Since the private verification algorithm begins by running the public one, both games are clearly identical until $\mathcal{A}$ queries the verification oracle on input of a candidate $(\boldsymbol{v}, (z, r, u, \pi_0), \mathsf{lbl})$ that is accepted by $\mathsf{V}(\psi, ., .)$ but rejected by $\mathsf{W}(\psi, \tau_v, ., .)$. If we call this event F_2, we have $|\Pr[S_1] - \Pr[S_2]| \leq \Pr[F_2]$.

Claim 1. The probability of event F_2 can be bounded as $\Pr[F_2] \leq \mathbf{Adv}^{\mathrm{SDP}}(\lambda) + \frac{1}{p}$.

Proof. We first remark that event F_2 can only occur for a candidate $\big(\boldsymbol{v}, (z, r, u, \pi_0), \mathsf{lbl}\big)$ such that the vector $(v_1, \ldots, v_n, \pi_0, v_1^\alpha, \ldots, v_n^\alpha)$ is not in the span of $\{\boldsymbol{H}_{2i-1}, \boldsymbol{H}_{2i}\}_{i=1}^t$. Indeed, otherwise, there would exist $\boldsymbol{x} = (x_1, \ldots, x_t) \in \mathbb{Z}_p^t$ such that $(v_1, \ldots, v_n) = g^{\boldsymbol{x} \cdot \mathbf{A}}$ and $\pi_0 = \prod_{i=1}^t (W_i^\alpha Y_i)^{x_i}$. In this case, we would also have

$$g^{\boldsymbol{x} \cdot \mathbf{A} \cdot (\boldsymbol{e}^\top + \alpha \cdot \boldsymbol{d}^\top)} = \prod_{i=1}^t (W_i^\alpha Y_i)^{x_i},$$

and the private verifier $\mathsf{W}(\psi, \tau_v, ., .)$ would accept the proof (z, r, u, π_0).

It comes that the only way for the adversary to cause a divergence between $\mathsf{W}(\psi, \tau_v, ., .)$ and $\mathsf{V}(\psi, ., .)$ is to create a valid-looking one-time linearly homomorphic signature (z, r, u) on a vector outside $\mathrm{span}\big(\{\boldsymbol{H}_{2i-1}, \boldsymbol{H}_{2i}\}_{i=1}^t\big)$. The result of [43][Theorem 1] shows that this occurs with probability at most $\Pr[F_2] \leq \mathbf{Adv}^{\mathrm{SDP}}(\lambda) + \frac{1}{p}$. ■

Game_3**:** This game like Game_2 but, when the adversary outputs its triple $(\boldsymbol{v}, \boldsymbol{x}, \mathsf{lbl})$, the challenger does not use the witness $\boldsymbol{x} \in \mathbb{Z}_p^t$ any longer. To simulate the proof for $(\boldsymbol{v}, \boldsymbol{x}, \mathsf{lbl})$, it first computes $\alpha = H(\rho, \boldsymbol{v}, \mathsf{lbl})$. Then, using the private vectors $\boldsymbol{d}, \boldsymbol{e} \in \mathbb{Z}_p^n$, it computes $\pi_0 = \prod_{j=1}^n v_j^{e_j+\alpha\cdot d_j}$ before using the private key $\mathsf{sk}_{ots} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^{2n+1}$ to compute a one-time signature

$$(z, r, u) = \Big(\prod_{j=1}^{2n+1} v_j^{-\chi_j}, \prod_{j=1}^{2n+1} v_j^{-\gamma_j}, \prod_{j=1}^{2n+1} v_j^{-\delta_j} \Big)$$

on the vector $\tilde{\boldsymbol{v}} = (v_1, \ldots, v_{2n+1})$, where $v_{n+1} = \pi_0$ and $v_{n+i+1} = v_i^{\alpha}$ for $i = 1$ to n. The resulting proof is easily seen to have the same distribution as in Game_2 when $\boldsymbol{v} \in \mathcal{L}_\rho$. We thus have $\Pr[S_3] = \Pr[S_2]$.

We define the simulator $(\mathsf{S}_1, \mathsf{S}_2)$ by having S_1 generate the CRS ψ as in Game_3 (observe that it has not changed since Game_1) and S_2 generate compute the proof without using the witnesses as in Game_3. The verification oracle is implemented as in Game_2 and Game_3. It easily comes that the system is computationally quasi-adaptive relatively zero-knowledge if the SDP assumption holds.

Quasi-Adaptive Relative Simulation-Soundness. We have to prove that, even if the simulator $(\mathsf{S}_1, \mathsf{S}_2)$ provides the adversary $\mathcal{A}$ with a simulated proof π for a pair $(\boldsymbol{v}, \mathsf{lbl})$, where $\boldsymbol{v} \in \mathbb{G}^n$ may not be in $\mathcal{L}_\rho$, $\mathcal{A}$ will remain unable to produce a new proof $(\boldsymbol{v}^\star, \pi^\star, \mathsf{lbl}^\star) \neq (\boldsymbol{v}, \pi, \mathsf{lbl})$ such that $\boldsymbol{v}^\star \notin \mathcal{L}_\rho$.

To prove the result, we rely on the smoothness of the projective hash function and on a specific property of the one-time linearly homomorphic signature of Section 2.5: namely, unless the SDP assumption is false, it is hard to compute two distinct signatures on the same vector, even when the private key is available.

We thus proceed with a sequence of games where the first game is the actual game and the last one is a game where the adversary has statistically no advantage. In Game_i, we denote by S_i the event that the adversary wins.

Game_1**:** is the real game where the adversary $\mathcal{A}$ is given the description of $\mathcal{L}_\rho$, a simulated CRS ψ and access to a simulated prover $\mathsf{S}_2(\psi, \tau_{sim}, ., .)$ which is queried only once. On this occasion, S_2 takes as input a vector $\boldsymbol{v}^\dagger \in \mathbb{G}^n$ and a label $\mathsf{lbl}^\dagger$ and it produces a simulated proof $\pi^\dagger = (z^\dagger, r^\dagger, u^\dagger, \pi_0^\dagger)$ that $\boldsymbol{v}^\dagger \in \mathcal{L}_\rho$. To generate $\rho \in \mathbb{G}^{t\times n}$ according to the distribution D_Γ at the beginning of the game, the challenger computes $\rho = g^{\mathbf{A}}$ after having sampled a matrix $\mathbf{A} \in \mathbb{Z}_p^{t\times n}$ with the appropriate distribution (which is possible since D_Γ is efficiently samplable). Moreover, the challenger $\mathcal{B}$ computes a basis $\mathbf{W} \in \mathbb{Z}_p^{n\times(n-t)}$ of the right kernel of $\mathbf{A}$. The adversary is allowed to query $\mathsf{S}_2(\psi, \tau_{sim}, ., .)$ exactly once and the private verifier $\mathsf{W}(\psi, \tau_v, ., .)$ on polynomially many occasions. When $\mathcal{A}$ terminates, it outputs a triple $(\boldsymbol{v}^\star, \pi^\star, \mathsf{lbl}^\star)$. The adversary is successful if $\boldsymbol{v}^\star \notin \mathcal{L}_\rho$ but $(\pi^\star, \mathsf{lbl}^\star)$ passes the private verification test and $(\boldsymbol{v}^\star, \pi^\star, \mathsf{lbl}^\star) \neq (\boldsymbol{v}^\dagger, \pi^\dagger, \mathsf{lbl}^\dagger)$. We denote by S_1 the latter event. Note that $\mathcal{B}$ can recognize this event as it can test if $\boldsymbol{v}^\star \notin \mathcal{L}_\rho$ by checking whether $\boldsymbol{v}^\star = (v_1^\star, \ldots, v_n^\star) \in \mathbb{G}^n$ satisfies $\prod_{j=1}^n v_j^{\star w_{ji}} = 1_{\mathbb{G}}$ for each column $\boldsymbol{w}_i^\top = (w_{1i}, \ldots, w_{ni})^\top$ of $\mathbf{W}$. Indeed, the vectors $\mathbf{y} \in \mathbb{Z}_p^n$ in the row space of $\mathbf{A}$ are exactly those for which $\mathbf{y} \cdot \mathbf{W} = \mathbf{0}$.

Game_2**:** In this game, we modify the behavior of the private verification oracle $\mathsf{W}(\psi, \tau_v, ., .)$. At each invocation (including the final invocation on the adversary's output $(\boldsymbol{v}^\star, \pi^\star, \mathsf{lbl}^\star)$) on input of a triple $(\boldsymbol{v}, \pi, \mathsf{lbl})$, the modified private verification oracle outputs 0 if $(\boldsymbol{v}, \pi, \mathsf{lbl}) \neq (\boldsymbol{v}^\dagger, \pi^\dagger, \mathsf{lbl}^\dagger)$ but $H(\rho, \boldsymbol{v}, \mathsf{lbl}) = H(\rho, \boldsymbol{v}^\dagger, \mathsf{lbl}^\dagger)$. Clearly, Game_1 and Game_2 proceed identically until the latter event, called F_2, occurs. We have $|\Pr[S_2] - \Pr[S_1]| \leq \Pr[F_2]$. Moreover, F_2 is unlikely to occur if H is a collision-resistant hash function: we have $\Pr[F_2] \leq \mathbf{Adv}^{\mathrm{CR}}(\lambda)$.

Game_3**:** This game is identical to Game_2 with the following difference. When the private verification oracle $\mathsf{W}(\psi, \tau_v, ., .)$ is run on input of the adversary's proof $(\boldsymbol{v}^\star, \pi^\star, \mathsf{lbl}^\star)$, it returns 0 in the event that $\pi^\star = (z^\star, r^\star, u^\star, \pi_0^\star)$ is such that $(z^\star, r^\star, u^\star) \neq (z^\dagger, r^\dagger, u^\dagger)$ but $(\boldsymbol{v}^\star, \mathsf{lbl}^\star) = (\boldsymbol{v}^\dagger, \mathsf{lbl}^\dagger)$. If we call F_3 the event that the private verification oracle $\mathsf{W}(\psi, \tau_v, ., .)$ rejects a proof that would have been accepted in Game_2, we have $|\Pr[S_3] - \Pr[S_2]| \leq \Pr[F_3]$. Moreover, F_3 implies a breach in the SDP assumption. Indeed, if $\pi_0^\star \neq \pi_0^\dagger$, $\mathsf{W}(\psi, \tau_v, ., .)$ would not accept $\pi^\star$ in Game_2 either, regardless of whether $(z^\star, r^\star, u^\star) \neq (z^\dagger, r^\dagger, u^\dagger)$ or not. If $\pi_0^\star = \pi_0^\dagger$, event F_3 provides the challenger with two distinct linearly homomorphic signatures $(z^\star, r^\star, u^\star)$ and $(z^\dagger, r^\dagger, u^\dagger)$ on the same vector $(v_1^\star, \ldots, v_n^\star, \pi_0^\star, v_1^{\star\alpha^\star}, \ldots, v_n^{\star\alpha^\star})$ as we also have $\alpha^\star = \alpha^\dagger$. As mentioned in Section 2.5 (and as can be easily observed from the verification equations), this would contradict the SDP assumption and we thus have $|\Pr[S_3] - \Pr[S_2]| \leq \mathbf{Adv}^{\mathrm{SDP}}(\lambda)$.

Game_4**:** In this game, we further modify the behavior of $\mathsf{W}(\psi, \tau_v, ., .)$ when it assesses the the adversary's output $(\boldsymbol{v}^\star, \pi^\star, \mathsf{lbl}^\star)$. Using the basis $\mathbf{W}$ of the right kernel of $\mathbf{A}$, the challenger $\mathcal{B}$ first checks if $\boldsymbol{v}^\star \notin \mathcal{L}_\rho$ and forces $\mathsf{W}(\psi, \tau_v, ., .)$ to return 0 whenever this is the case. If we denote by F_4 the event that $\mathsf{W}(\psi, \tau_v, ., .)$ rejects an adversarially-generated triple $(\boldsymbol{v}^\star, \pi^\star, \mathsf{lbl}^\star)$ that would have survived the private verification test of Game_3, we have $|\Pr[S_4] - \Pr[S_3]| \leq \Pr[F_4]$. Since $\Pr[S_4] = 0$ by construction, we are left with the task of bounding $\Pr[F_4]$.

Claim 2. The probability of event F_4 is at most $\Pr[F_4] \leq 1/(p-q)$, where q denotes the number of private verification queries.

Proof. The proof of the claim is a standard argument based on the smoothness of the projective hash function. If we consider the information that $\mathcal{A}$ can infer about the private evaluation key $\tau_v = \big(\boldsymbol{d} = (d_1, \ldots, d_n), \boldsymbol{e} = (e_1, \ldots, e_n)\big)$ by observing the CRS and the proof $\pi^\dagger = (z^\dagger, r^\dagger, u^\dagger, \pi_0^\dagger)$, it amounts to the first $2t+1$ rows of the left-hand-side member of the following linear system:

$$\begin{pmatrix} \mathbf{Y}^\top \\ \mathbf{W}^\top \\ \pi_0^\dagger \\ \pi_0^\star \end{pmatrix} = \begin{pmatrix} \mathbf{A} & \\ & \mathbf{A} \\ \log(\boldsymbol{v}^\dagger) & \alpha^\dagger \cdot \log(\boldsymbol{v}^\dagger) \\ \log(\boldsymbol{v}^\star) & \alpha^\star \cdot \log(\boldsymbol{v}^\star) \end{pmatrix} \cdot \begin{pmatrix} \mathbf{e}^\top \\ \mathbf{d}^\top \end{pmatrix} \tag{13}$$

Let us assume that $\boldsymbol{v}^\dagger, \boldsymbol{v}^\star \notin \mathcal{L}_\rho$. Since the above $(2t+2) \times 2n$ matrix has full rank when $\alpha^\star \neq \alpha$ (which is the case unless the failure event F_1 of Game_1 occurs), we see that the only value of $\pi_0^\star$ that would trick the private verification oracle $\mathsf{W}(\psi, \tau, ., .)$ of Game_3 into accepting $\pi^\star$ is completely independent of the information provided by the CRS and the simulated proof $\pi^\dagger$ for $\boldsymbol{v}^\dagger$. However, $\mathcal{A}$ can also take advantage of its private verification queries throughout the game. Without any verification query, $(\boldsymbol{e}, \boldsymbol{d})$ is constrained by the first $2t+1$ rows of (13) to live in a subspace of dimension $2(n-t)-1 \geq 1$ in $\mathbb{Z}_p^{2n}$, so that $\pi_0^\star$ has p equally likely values in $\mathcal{A}$'s view. Each private verification query provides $\mathcal{A}$ with an inequality, which allows it to rule out one candidate for the value of $\pi_0^\star$ that $\mathsf{W}(\psi, \tau_v, ., .)$ would accept. After q queries, $\mathcal{A}$ is thus left with $p - q$ equally likely candidates for $\pi_0^\star$. We thus find $\Pr[F_4] \leq 1/(p-q)$, as claimed. ■

Putting the above altogether, $\mathcal{A}$'s advantage is breaking the quasi-adaptive relative simulation-soundness property is bounded as

$$\mathbf{Adv}^{\mathrm{rss}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{CR}}(\lambda) + \mathbf{Adv}^{\mathrm{SDP}}(\lambda) + \frac{1}{2^\lambda - q}.$$

□

H Proof of KH-CCA Security for the Keyed-Homomorphic Scheme

Instead of relying on the simulation-soundness of the proof system in a modular manner, our proof of KH-CCA security uses a direct security analysis in order to obtain a tighter reduction.

Theorem 3. *The threshold keyed-homomorphic cryptosystem of Section 5 provides KH-CCA security in the sense of Definition 1 assuming that: (i) Σ is a strongly unforgeable one-time signature; (ii) The DLIN assumption holds in $\mathbb{G}$. Concretely, the advantage of any PPT adversary $\mathcal{A}$ is at most*

$$\mathbf{Adv}^{\text{kh-cca}}(\mathcal{A}) < (q_e+1)\cdot\mathbf{Adv}^{\text{suf-ots}}(\lambda) + \frac{3}{2}\cdot\mathbf{Adv}^{\text{DLIN}}(\lambda) + 7\cdot\sqrt{(q_e+1)\cdot(L+1)\cdot\mathbf{Adv}^{\text{DLIN}}(\lambda)\cdot\big(1-\frac{1}{p}\big)^{-1}} + \frac{1}{p},$$

where $\mathbf{Adv}^{\text{suf-ots}}(\lambda)$ denotes $\mathcal{A}$'s probability to break the strong unforgeability of Σ and q_e is the maximal number of evaluation queries involving derivatives of the challenge ciphertext.

Proof. The proof uses of a sequence of games starting with the real attack game and ending with a game where the adversary $\mathcal{A}$ has no advantage. For each i, we also denote by S_i the event that the challenger outputs 1 in Game_i.

Game_1**:** is the actual attack game with the only difference that the challenger $\mathcal{B}$ does not erase sk_{ot} during the key generation phase. In details, the adversary is given the public key PK and the set of verification keys $\mathbf{VK} = (VK_1, \ldots, VK_N)$. If $\mathcal{A}$ decides to query the RevHK oracle at some point, $\mathcal{B}$ reveals SK_h. At each corruption query $i \in \{1, \ldots, N\}$, $\mathcal{B}$ reveals the queried private key share $SK_i = (P_1(i), P_2(i), P(i))$. At each decryption query, $\mathcal{B}$ faithfully runs the real shared decryption algorithm. At each evaluation query, $\mathcal{A}$ supplies two ciphertexts $C^{(1)}, C^{(2)}$ which are processed by $\mathcal{B}$ as in the evaluation algorithm. We denote by $C_1^\dagger, \ldots, C_{q_e}^\dagger$ the outputs of the $\mathsf{Eval}(SK_h, .)$ oracle when the latter is queried on a pair $(C^{(1)}, C^{(2)})$ such that $C^{(j)} \in \mathcal{D}$ for some $j \in \{1, 2\}$ (in other words, $\{C_l^\dagger\}_{l=1}^{q_e}$ are the results of evaluation queries which increase $|\mathcal{D}|$). We also denote by $\mathsf{SVK}_1^\dagger, \ldots, \mathsf{SVK}_{q_e}^\dagger$ the one-time verification keys appearing in these ciphertexts and assume w.l.o.g. that they are chosen by $\mathcal{B}$ at the very beginning of the game.

When the first phase is over, the adversary $\mathcal{A}$ chooses two distinct messages $M_0, M_1 \in \mathbb{G}$ and obtains $C^\star = (\mathsf{SVK}^\star, C_0^\star, C_1^\star, C_2^\star, C_3^\star, Z^\star, R^\star, U^\star, \boldsymbol{C}_z^\star, \boldsymbol{C}_r^\star, \boldsymbol{C}_u^\star, \boldsymbol{\pi}_1^\star, \boldsymbol{\pi}_2^\star, \sigma^\star)$ which is an encryption of M_β, for some random coin $\beta \stackrel{R}{\leftarrow} \{0, 1\}$ flipped by $\mathcal{B}$.

In the second phase, $\mathcal{A}$ makes more decryption, evaluation and corruption queries under the restriction of not asking for a partial decryption of a ciphertext in $\mathcal{D}$ or for more than $t-1$ private key shares throughout the entire game. Eventually, $\mathcal{A}$ halts and outputs $\beta' \in \{0, 1\}$. The challenger $\mathcal{B}$ outputs 1 if and only if $\beta = \beta'$. We denote this event by S_1.

Game_2**:** This game is identical to Game_1 with the difference that the challenger $\mathcal{B}$ rejects all decryption queries involving ciphertexts $C = (\mathsf{SVK}, C_0, C_1, C_2, C_3, Z, R, U, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \sigma)$ such that $\mathsf{SVK} \in \{\mathsf{SVK}^\star, \mathsf{SVK}_1^\dagger, \ldots, \mathsf{SVK}_{q_e}^\dagger\}$. It also returns $\perp$ at each evaluation query $(C^{(1)}, C^{(2)})$ for which there exists $j \in \{1, 2\}$ for which $C^{(j)}$ contains a verification key $\mathsf{SVK}^{(j)}$ such that $\mathsf{SVK}^{(j)} = \mathsf{SVK}_l^\dagger$, for some $\mathsf{SVK}_l^\dagger \in \{\mathsf{SVK}^\star, \mathsf{SVK}_1^\dagger, \ldots, \mathsf{SVK}_{q_e}^\dagger\}$, but $C^{(j)} \neq C_l^\dagger$.

If we define F_2 to be the event that $\mathcal{B}$ rejects a ciphertext that would not have been rejected in Game_1, we see that Game_2 and Game_1 proceed identically until event F_2 occurs. We thus have the inequality $|\Pr[S_2] - \Pr[S_1]| \leq \Pr[F_2]$. Moreover, event F_2 would imply a breach in the strong unforgeability of the one-time signature. Indeed, since $\mathcal{A}$ is not allowed to query the partial decryption of any ciphertext in $\mathcal{D}$, a standard argument allows proving the inequality $|\Pr[S_2] - \Pr[S_1]| \leq \Pr[F_2] \leq (q_e + 1) \cdot \mathbf{Adv}^{\text{suf-ots}}(\lambda)$. In the forthcoming games, we will assume that $\mathsf{SVK} \notin \{\mathsf{SVK}^\star, \mathsf{SVK}_1^\dagger, \ldots, \mathsf{SVK}_{q_e}^\dagger\}$ at each decryption query.

Game_3**:** We modify the generation of the challenge ciphertext $C^\star$. Namely, instead of computing $C_0^\star = M_\beta \cdot X_1^\theta \cdot X_2^\theta$, using the encryption exponents $\theta_1 = \log_f(C_1^\star)$ and $\theta_2 = \log_h(C_2^\star)$, $\mathcal{B}$ uses the private key (x_0, x_1, x_2) and computes $C_0^\star = M_\beta \cdot C_1^{\star x_1} \cdot C_2^{\star x_2} \cdot C_3^{\star x_0}$. Likewise, instead of using

(θ_1, θ_2) to derive the triple $(z^\star, r^\star, u^\star)$ at step 3 of the encryption algorithm, $\mathcal{B}$ uses the simulation trapdoor $\mathsf{sk}_{rand} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^{3}$ of the USS proof system and computes

$$z^\star = \prod_{i=1}^{3} C_i^{\star-\chi_i} \qquad r^\star = \prod_{i=1}^{3} C_i^{\star-\gamma_i} \qquad u^\star = \prod_{i=1}^{3} C_i^{\star-\delta_i}. \tag{14}$$

Finally, $(Z^\star, R^\star, U^\star)$ is generated using $\mathsf{sk}_{ot} = \{(\varphi_i, \vartheta_i, \varpi_i)\}_{i=1}^{3}$ as

$$Z^\star = \prod_{i=1}^{3} C_i^{\star-\varphi_i} \qquad R^\star = \prod_{i=1}^{3} C_i^{\star-\vartheta_i} \qquad U^\star = \prod_{i=1}^{3} C_i^{\star-\varpi_i}.$$

Then, $\mathcal{B}$ conducts steps 4-6 as in the actual encryption algorithm. It is easy to see that this change does not modify $\mathcal{A}$'s view since $C_0^\star$ still equals $C_0^\star = M_\beta \cdot X_1^{\theta_1} \cdot X_2^{\theta_2}$ and the distributions of $(z^\star, r^\star, u^\star)$ and $(Z^\star, R^\star, U^\star)$ remain the same as in Game_2. We thus have $\Pr[S_3] = \Pr[S_2]$.

Game_4**:** This game is identical to Game_3 with a new modification in the challenge ciphertext. Namely, instead of setting $C_3^\star = g^{\theta_1+\theta_2}$, where $\theta_1 = \log_f(C_1^\star)$ and $\theta_2 = \log_h(C_2^\star)$, we choose it as $C_3^\star \xleftarrow{R} \mathbb{G}$. At the third step of the encryption algorithm, the linearly homomorphic signature $(z^\star, r^\star, u^\star)$ is computed according to (14), as previously. Under the DLIN assumption in $\mathbb{G}$, this modification should not significantly alter $\mathcal{A}$'s behavior. In particular, we have $|\Pr[S_4] - \Pr[S_3]| \leq \mathbf{Adv}^{\mathrm{DLIN}}(\lambda)$.

Game_5**:** From this point forward, we make explicit use of the discrete logarithms $\alpha_f = \log_g(f)$ and $\alpha_h = \log_g(h)$, which is allowed since we are done with the transition consisting in tampering with $C_3^\star$ in the challenge ciphertext. In Game_5, we first change the treatment of ciphertexts $C = (\mathsf{SVK}, C_0, C_1, C_2, C_3, Z, R, U, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \sigma)$ involved in *pre-challenge* decryption and evaluation queries. Namely, $\mathcal{B}$ simply ignores the linearly homomorphic signatures contained these ciphertexts and returns $\perp$ if $C_3 \neq C_1^{1/\alpha_f} \cdot C_2^{1/\alpha_h}$. Otherwise, it responds as in earlier games.

If we call F_5 the event that $\mathcal{B}$ rejects a ciphertext that would not have been rejected in Game_4, Game_5 and Game_4 are clearly identical until F_5 occurs, so that $|\Pr[S_5] - \Pr[S_4]| \leq \Pr[F_5]$. At the same time, Lemma 1 shows that $\Pr[F_5] \leq \frac{1}{2} \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\lambda) + 1/p$.

The proof of Lemma 1 implies that, even if $\mathcal{A}$ chooses to expose the evaluation key SK_h at the very beginning of the game, it should not be able to create valid-looking ill-formed ciphertexts before the challenge phase unless the DLIN assumption is false. This will help us prove that the scheme remains IND-CCA1 if SK_h is revealed to the adversary when the game begins.

Game_6**:** We now modify the treatment of *post-challenge* queries and introduce yet another event F_6. In this game, the challenger $\mathcal{B}$ halts and outputs a random bit in the event that the adversary $\mathcal{A}$ manages to query the partial decryption oracle or the evaluation oracle on a ciphertext $C^\diamond = (\mathsf{SVK}^\diamond, C_0^\diamond, C_1^\diamond, C_2^\diamond, C_3^\diamond, Z^\diamond, R^\diamond, U^\diamond, \boldsymbol{C}_z^\diamond, \boldsymbol{C}_r^\diamond, \boldsymbol{C}_u^\diamond, \boldsymbol{\pi}_1^\diamond, \boldsymbol{\pi}_2^\diamond, \sigma^\diamond)$ where $\mathsf{SVK}^\diamond \not\in \{\mathsf{SVK}^\star, \mathsf{SVK}_1^\dagger, \ldots, \mathsf{SVK}_{q_e}^\dagger\}$ and $C_3^\diamond \neq C_1^{\diamond 1/\alpha_f} \cdot C_2^{\diamond 1/\alpha_h}$ although $(\boldsymbol{C}_z^\diamond, \boldsymbol{C}_r^\diamond, \boldsymbol{C}_u^\diamond, \boldsymbol{\pi}_1^\diamond, \boldsymbol{\pi}_2^\diamond)$ is a valid linearly homomorphic signature on the vector $(C_1^\diamond, C_2^\diamond, C_3^\diamond)$. We say that $C^\diamond$ is a *fatal* query in this case. Since Game_6 is identical to Game_5 until event F_6 occurs, we have $|\Pr[S_6] - \Pr[S_5]| \leq \Pr[F_6]$. Lemma 2 demonstrates that the DLIN assumption can be broken in the group $\mathbb{G}$ if event F_6 occurs with non-negligible probability. More precisely, Lemma 2 shows that $\Pr[F_6] \leq 7 \cdot \sqrt{(q_e + 1) \cdot (L + 1) \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\lambda) \cdot \left(1 - \frac{1}{p}\right)^{-1}}$.

Game_7**:** We now modify the partial decryption oracle and replace the non-interactive proofs contained in decryption shares $\hat{\mu_i}$ by simulated NIZK proofs. This entails to turn $(\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3)$ into a perfectly hiding Groth-Sahai CRS (where $\boldsymbol{f}_3$ is in the span of $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$) and non-interactive proofs for multi-exponentiation equations are simulated using the trapdoor of the simulated CRS. Under the DLIN assumption, this change is not noticeable to $\mathcal{A}$ and we have $|\Pr[S_7] - \Pr[S_6]| \leq \mathbf{Adv}^{\mathrm{DLIN}}(\lambda)$.

In Game_7, it is easy to see that $\mathcal{A}$ has no advantage whatsoever, so that $\Pr[S_7] = 1/2$. Indeed, in the challenge phase, we have $(C_1^\star, C_2^\star, C_3^\star) = (f^{\theta_1}, h^{\theta_2}, g^{\theta_1+\theta_2+\theta_3})$, where $\theta_1, \theta_2, \theta_3 \in_R \mathbb{Z}_p$. This implies that $C_0^\star$ can be written as $C_0^\star = M_\beta \cdot X_1^{\theta_1} \cdot X_2^{\theta_2} \cdot g^{\theta_3 \cdot x_0}$. The latter equality implies that, as long as $x_0 \in \mathbb{Z}_p$ remains independent of $\mathcal{A}$'s view, so does the challenger's bit $\beta \in \{0, 1\}$.

To see why $\mathcal{A}$ does not learn anything about $x_0 \in \mathbb{Z}_p$, we first note that the homomorphic evaluation key SK_h is independent of x_0, so that homomorphic evaluation queries leak nothing about it. We also remark that, in Game_7, decryption shares $\hat{\mu_i}$ contain NIZK proofs that are simulated without using private key shares. Hence, as long as $\mathcal{A}$ does not corrupt more than $t-1$ servers, the only possible way to infer information about $x_0 = P(0)$ is to trick the partial decryption oracle into accepting an invalid ciphertext. However, in Game_7, all invalid ciphertexts are explicitly rejected.

We thus find the announced result

$$
\begin{aligned}
|\Pr[S_1] - \frac{1}{2}| < (q_e + 1) \cdot \mathbf{Adv}^{\text{suf-ots}}(\lambda) + \frac{3}{2} \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\lambda) \\
+ 7 \cdot \sqrt{(q_e + 1) \cdot (L + 1) \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\lambda) \cdot \big(1 - \frac{1}{p}\big)^{-1}} + \frac{1}{p}.
\end{aligned}
$$

□

Lemma 1. *In* Game_5*, the probability of event* F_5 *is at most* $\Pr[F_5] \leq \frac{1}{2} \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\lambda) + \frac{1}{p}$.

Proof. We show that, if event F_5 occurs with non-negligible probability in Game_5, there exists an efficient algorithm $\mathcal{B}$ that solves a SDP instance (G_z, G_r, H_z, H_u) with nearly the same probability. In turn, $\mathcal{B}$ implies a distinguisher for the DLIN assumption in $\mathbb{G}$.

Algorithm $\mathcal{B}$ generates public key components are defined as in the actual scheme. In particular, $\mathcal{B}$ sets $G_i = G_z{}^{\varphi_i} G_r{}^{\vartheta_i}$ and $H_i = H_z{}^{\varphi_i} H_u{}^{\varpi_i}$ with $\varphi_i, \vartheta_i, \varpi_i \xleftarrow{R} \mathbb{Z}_p$ for $i \in \{1, 2, 3\}$.

Throughout the game, the reduction $\mathcal{B}$ answers $\mathcal{A}$'s decryption queries in the same way as in Game_5. Moreover, since $\mathcal{B}$ has generated $(\mathsf{sk}_{rand}, \mathsf{pk}_{rand})$ faithfully, it is able to consistently reveal the evaluation key SK_h in case $\mathcal{A}$ decides to corrupt the evaluator. If event F_5 occurs with non-negligible probability, we know that, before the challenge phase, $\mathcal{A}$ must query the partial decryption or the homomorphic evaluation of a ciphertext $C = (\mathsf{SVK}, C_0, C_1, C_2, C_3, Z, R, U, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \sigma)$ such that (Z, R, U) is a valid one-time linearly homomorphic signature on (C_1, C_2, C_3) although (C_1, C_2, C_3) is outside the span of $(f, 1, g)$ and $(1, h, g)$. When algorithm $\mathcal{B}$ detects this event (by observing that $C_3 \neq C_1^{1/\alpha_f} C_2^{1/\alpha_h}$), it computes its own signature

$$
(Z^\dagger, R^\dagger, U^\dagger) = (\prod_{i=1}^{3} C_i{}^{-\varphi_i}, \prod_{i=1}^{3} C_i{}^{-\vartheta_i}, \prod_{i=1}^{3} C_i{}^{-\varpi_i}) \tag{15}
$$

on (C_1, C_2, C_3). We claim that, with overwhelming probability,

$$
(Z^\ddagger, R^\ddagger, U^\ddagger) = \Big(\frac{Z}{Z^\dagger}, \frac{R}{R^\dagger}, \frac{U}{U^\dagger}\Big)
$$

is a non-trivial solution to the SDP instance since $Z^\ddagger \neq 1_{\mathbb{G}}$ with overwhelming probability.

Indeed, we remark that the vector $(\varphi_1, \varphi_2, \varphi_3)$ is independent of $\mathcal{A}$'s view before the challenge phase. Consequently, since (C_1, C_2, C_3) is linearly independent of $(f, 1, g)$ and $(1, h, g)$, the adversary $\mathcal{A}$ is only able to predict the value $Z^\dagger$ of (15) with probability $1/p$. Given that we also have the inequality $\mathbf{Adv}^{\mathrm{SDP}}(\lambda) \leq \frac{1}{2} \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\lambda)$, we thus find $\Pr[F_5] \leq \frac{1}{2} \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\lambda) + \frac{1}{p}$ as claimed. □

In the transition from Game_5 to Game_6, we could rely on the unbounded simulation-soundness of the underlying QA-NIZK proof to argue that fatal decryption or evaluation queries only occur with negligible probability. To do this, we would have to build a reduction algorithm that interacts with a simulation-soundness challenger for a given matrix $\rho \in \mathbb{G}^{2 \times 3}$ and simultaneously emulates $\mathcal{A}$'s challenger in the KH-CCA game. Since the reduction would not have the matrix $\mathbf{A} \in \mathbb{Z}_p^{2 \times 3}$, it would have no way to detect fatal queries. Consequently, the reduction would have to guess this query, which would introduce an extra degradation factor in the reduction.

Lemma 2. *In* Game_6*, the probability of event* F_6 *is at most*

$$\Pr[F_6] \leq 7 \cdot \sqrt{(q_e+1)\cdot(L+1)\cdot \mathbf{Adv}^{\mathrm{DLIN}}(\lambda)\cdot \big(1-\frac{1}{p}\big)^{-1}}.$$

Proof. The key argument of the proof is that, conditionally on a certain desirable event, the evaluation oracle $\mathsf{Eval}(SK_h, .)$ will never information-theoretically reveal its evaluation key SK_h.

Assuming that event F_6 occurs with non-negligible probability in Game_6, we show that there exists a distinguisher for the DLIN assumption in $\mathbb{G}$. To this end, we consider a subsequence of games starting with Game_6 and ending with $\mathsf{Game}_{6.2}$. For each $j \in \{0,1,2\}$, we define $F_{6.j}$ as the counterpart of event F_6 in $\mathsf{Game}_{6.j}$ (note that $F_{6.j}$ is efficiently detectable for each j). We first show that, as long as the DLIN assumption holds, if $\Pr[F_6]$ is non-negligible, so is $\Pr[F_{6.2}]$.

$\mathsf{Game}_{6.0}$: This game is identical to Game_6 but we modify the generation of pk_{rand} when the public key is set up. Namely, the vectors $(\boldsymbol{f}_1, \boldsymbol{f}_2, \{\boldsymbol{f}_{3,i}\}_{i=0}^{L})$ are chosen by setting $\boldsymbol{f}_1 = (f_1, 1_{\mathbb{G}}, g)$ and $\boldsymbol{f}_2 = (1_{\mathbb{G}}, f_2, g)$ where $f_1, f_2 \xleftarrow{R} \mathbb{G}$ are chosen at random. As for $\{\boldsymbol{f}_{3,i}\}_{i=0}^{L}$, they are obtained as

$$\begin{aligned} \boldsymbol{f}_{3,0} &= \boldsymbol{f}_1^{\xi_{0,1}} \cdot \boldsymbol{f}_2^{\xi_{0,2}} \cdot (1,1,g)^{\xi_{0,3}} \cdot (1,1,g)^{\mu\cdot\zeta-\rho_0} && (16)\\ \boldsymbol{f}_{3,i} &= \boldsymbol{f}_1^{\xi_{i,1}} \cdot \boldsymbol{f}_2^{\xi_{i,2}} \cdot (1,1,g)^{\xi_{i,3}} \cdot (1,1,g)^{-\rho_i}, && i \in \{1,\ldots,L\} \end{aligned}$$

with $\mu \xleftarrow{R} \{0,\ldots,L\}$, $\xi_{0,1}, \xi_{1,1}, \ldots, \xi_{L,1} \xleftarrow{R} \mathbb{Z}_p$, $\xi_{0,2}, \xi_{1,2}, \ldots, \xi_{L,2} \xleftarrow{R} \mathbb{Z}_p$, $\xi_{0,3}, \xi_{1,3}, \ldots, \xi_{L,3} \xleftarrow{R} \mathbb{Z}_p$ and $\rho_0, \rho_1, \ldots, \rho_L \xleftarrow{R} \{0,\ldots,\zeta-1\}$, with $\zeta = 2(q_e+1)$ and where q_e is the number of evaluation queries that increase the cardinality of $\mathcal{D}$. Note that this change is only conceptual since the distribution of $\{\boldsymbol{f}_{3,i}\}_{i=0}^{L}$ has not changed since Game_6. We thus have $\Pr[F_{6.0}] = \Pr[F_6]$.

$\mathsf{Game}_{6.1}$: This game is like $\mathsf{Game}_{6.0}$ but we consider another event Good which causes the challenger $\mathcal{B}$ to abort and output a random bit if it does *not* occur. Let $\mathsf{SVK}_1^{\dagger}, \ldots, \mathsf{SVK}_{q_e}^{\dagger}$ be the distinct one-time verification keys appearing in outputs of the $\mathsf{Eval}(SK_h, .)$ oracle when the latter is invoked on a ciphertext in $\mathcal{D}$. Let also $\mathsf{SVK}^{\star}$ be the verification key involved in the challenge ciphertext $C^{\star}$ and let $\mathsf{SVK}^{\diamond}$ be the one involved in the first fatal query $C^{\diamond}$. We know that $\mathsf{SVK}^{\diamond} \notin \{\mathsf{SVK}^{\star}, \mathsf{SVK}_1^{\dagger}, \ldots, \mathsf{SVK}_{q_e}^{\dagger}\}$. For each verification key $\mathsf{SVK} \in \{0,1\}^L$, we consider the function $J(\mathsf{SVK}) = \mu\cdot\zeta - \rho_0 - \sum_{i=1}^{L} \rho_i \cdot \mathsf{SVK}[i]$. We also define Good to be the event that

$$J(\mathsf{SVK}^{\diamond}) = 0 \qquad \wedge \qquad J(\mathsf{SVK}^{\star}) \neq 0 \qquad \wedge \qquad \bigwedge_{j\in\{1,\ldots,q_e\}} J(\mathsf{SVK}_j^{\dagger}) \neq 0. \tag{17}$$

We remark that the random exponents $\rho_0, \rho_1, \ldots, \rho_L$ are chosen independently of $\mathcal{A}$'s view: this means that the simulator could equivalently define $\{\boldsymbol{f}_{3,i}\}_{i=0}^{L}$ first and only choose $\{\rho_i\}_{i=0}^{L}$ – together with values $\{\xi_{3,i}\}_{i=0}^{L}$ explaining the $\{\boldsymbol{f}_{3,i}\}_{i=0}^{L}$ – at the very end of the game, when $\mathsf{SVK}^{\star}, \mathsf{SVK}_1^{\dagger}, \ldots, \mathsf{SVK}_{q_e}^{\dagger}, \mathsf{SVK}^{\diamond}$ have been defined. The same analysis as [54] (using the simplifications of Bellare and Ristenpart [6]) shows that $\Pr[F_{6.1} \wedge \mathsf{Good}] \geq \Pr[F_{6.0}]^2/(27\cdot(q_e+1)\cdot(L+1))$.

This follows from the fact that, for any set of queries, a lower bound on the probability of event Good is $1/(2(q_e+1)(L+1))$. Indeed, from the known results [54, 35] on the programmability of Waters' hash function, we know that the probability, taken over the choice of $(\mu, \rho_0, \ldots, \rho_L)$, to meet the conditions (17) is at least $1/(2(q_e+1)(L+1))$.

$\mathsf{Game}_{6.2}$: We modify again the way to compute pk_{rand} in the generation of the public key. Namely, the vectors $\boldsymbol{f_1} = (f_1, 1_{\mathbb{G}}, g)$, $\boldsymbol{f_2} = (1_{\mathbb{G}}, f_2, g)$ are chosen as before. However, instead of generating $\{\boldsymbol{f}_{3,i}\}_{i=0}^{L}$ as in $\mathsf{Game}_{6.1}$, we set them as

$$\begin{aligned} \boldsymbol{f}_{3,0} &= \boldsymbol{f}_1^{\xi_{0,1}} \cdot \boldsymbol{f}_2^{\xi_{0,2}} \cdot (1,1,g)^{\mu\cdot\zeta-\rho_0} && (18)\\ \boldsymbol{f}_{3,i} &= \boldsymbol{f}_1^{\xi_{i,1}} \cdot \boldsymbol{f}_2^{\xi_{i,2}} \cdot (1,1,g)^{-\rho_i}, && i \in \{1,\ldots,L\} \end{aligned}$$

which amounts to setting $\xi_{0,3} = \xi_{1,3} = \ldots = \xi_{L,3} = 0$. Clearly, $\{\boldsymbol{f}_{3,i}\}_{i=0}^{L}$ are no longer uniform in the span of $\big(\boldsymbol{f}_1, \boldsymbol{f}_2, (1,1,g)\big)$. Still, this change should have no noticeable effect on $\mathcal{A}$ if the DLIN assumption holds in $\mathbb{G}$. Concretely, if a fatal decryption/evaluation query occurs with substantially different probabilities in $\mathsf{Game}_{6.2}$ and $\mathsf{Game}_{6.1}$, we can construct a DLIN distinguisher $\mathcal{B}^{\mathrm{DLIN}}$ in the group $\mathbb{G}$ (recall that the reduction can detect fatal queries using $\alpha_f = \log_g(f)$ and $\alpha_h = \log_g(h)$). This distinguisher uses the random self-reducibility of DLIN to construct many independent-looking instances from the same distribution out of a given instance. For this reason, we can write $|\Pr[F_{6.2} \wedge \mathsf{Good}] - \Pr[F_{6.1} \wedge \mathsf{Good}]| \leq \mathbf{Adv}_{\mathcal{B}^{\mathrm{DLIN}}}(\lambda)$.

In $\mathsf{Game}_{6.2}$, we show that an occurrence of event $F_{6.2}$ implies an algorithm $\mathcal{B}$ solving a given SDP instance (g_z, g_r, h_z, h_u) with non-negligible probability, which *a fortiori* breaks the DLIN assumption in $\mathbb{G}$ as the latter is implied by SDP.

By hypothesis, we know that the adversary $\mathcal{A}$ somehow manages to produce a fatal decryption/evaluation query on a ciphertext $C^\diamond$ for which $(C_1^\diamond, C_2^\diamond, C_3^\diamond)$ is outside the span of $(f, 1_{\mathbb{G}}, g)$ and $(1_{\mathbb{G}}, h, g)$ but $(\boldsymbol{C}_z^\diamond, \boldsymbol{C}_r^\diamond, \boldsymbol{C}_u^\diamond, \boldsymbol{\pi}_1^\diamond, \boldsymbol{\pi}_2^\diamond) \in \mathbb{G}^{15}$ satisfies the verification equations. At this point, if the event Good introduced in $\mathsf{Game}_{6.1}$ occurs, we must have $J(\mathsf{SVK}^\diamond) = 0$, which implies that $\boldsymbol{f}_{\mathsf{SVK}^\diamond} = \boldsymbol{f}_{3,0} \cdot \prod_{i=1}^{L+1} \boldsymbol{f}_{3,i}^{\mathsf{SVK}^\diamond[i]}$ lies in $\mathrm{span}(\boldsymbol{f}_1, \boldsymbol{f}_2)$. Consequently, $\boldsymbol{C}_z^\diamond$, $\boldsymbol{C}_r^\diamond$ and $\boldsymbol{C}_u^\diamond$ are necessarily perfectly binding and extractable commitments. Using $(\log_g(f_1), \log_g(f_2))$, $\mathcal{B}$ can thus extract the committed group elements $(z^\diamond, r^\diamond, u^\diamond) \in \mathbb{G}^3$ by BBS-decrypting the ciphertexts $(\boldsymbol{C}_z^\diamond, \boldsymbol{C}_r^\diamond, \boldsymbol{C}_u^\diamond)$. Since $(\boldsymbol{\pi}_1^\diamond, \boldsymbol{\pi}_2^\diamond)$ are perfectly sound Groth-Sahai proofs, the extracted elements $(z^\diamond, r^\diamond, u^\diamond)$ necessarily satisfy

$$1_{\mathbb{G}_T} = e(g_z, z^\diamond) \cdot e(g_r, r^\diamond) \cdot \prod_{i=1}^{3} e(g_i, C_i^\diamond) = e(h_z, z^\diamond) \cdot e(h_u, u^\diamond) \cdot \prod_{i=1}^{3} e(h_i, C_i^\diamond). \tag{19}$$

Having extracted $(z^\diamond, r^\diamond, u^\diamond)$, $\mathcal{B}$ also computes

$$z^\dagger = \prod_{i=1}^{3} C_i^{\diamond -\chi_i} \qquad r^\dagger = \prod_{i=1}^{3} C_i^{\diamond -\gamma_i} \qquad u^\dagger = \prod_{i=1}^{3} C_i^{\diamond -\delta_i}, \tag{20}$$

so that $(z^\dagger, r^\dagger, u^\dagger)$ also satisfies (19). Since $(z^\dagger, r^\dagger, u^\dagger)$ and $(z^\diamond, r^\diamond, u^\diamond)$ both satisfy (19), the triple

$$(z^\ddagger, r^\ddagger, u^\ddagger) = \Big(\frac{z^\diamond}{z^\dagger}, \frac{r^\diamond}{r^\dagger}, \frac{u^\diamond}{u^\dagger}\Big)$$

satisfies $e(g_z, z^\ddagger) \cdot e(g_r, r^\ddagger) = e(h_z, z^\ddagger) \cdot e(h_u, u^\ddagger) = 1_{\mathbb{G}_T}$. To conclude the proof, we argue that $z^\ddagger \neq 1_{\mathbb{G}}$ with overwhelming probability.

To do this, we observe that, if the event Good defined in $\mathsf{Game}_{6.1}$ actually comes about, then $\mathcal{B}$ never leaks any more information about (χ_1, χ_2, χ_3) than $\mathcal{A}$ can infer by just observing $\{(z_j, r_j, u_j)\}_{j=1}^{2}$ in the public key. Indeed, in this case we have $J(\mathsf{SVK}^\star) \neq 0$ and $J(\mathsf{SVK}_j^\dagger) \neq 0$ for each $j \in \{1, \ldots, q_e\}$. This means that, in the challenge ciphertext and all its homomorphic evaluations, the proofs $(\boldsymbol{\pi_1}, \boldsymbol{\pi_2})$ are perfectly WI as they are generated for a perfectly hiding Groth-Sahai CRS. For these ciphertexts, the built-in homomorphic signatures $(\boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi_2})$ leak nothing about the specific vector $(\chi_1, \chi_2, \chi_3) \in \mathbb{Z}_p^3$ used by $\mathcal{B}$. As a consequence, we can apply the same arguments as in the proof of Lemma 1 when it comes to argue that $z^\dagger \neq z^\diamond$ with probability $1 - 1/p$. We thus find

$$\Pr[F_{6.2} \wedge \mathsf{Good}] = \mathbf{Adv}_{\mathcal{B}}^{\mathrm{SDP}}(\lambda) \cdot \Big(1 - \frac{1}{p}\Big)^{-1}.$$

In turn, $\mathcal{B}$ implies a PPT distinguisher $\mathcal{B}^{\mathrm{DLIN}'}$ for the DLIN assumption such that we have the inequality $\Pr[F_{6.2} \wedge \mathsf{Good}] < \frac{1}{2} \cdot \mathbf{Adv}_{\mathcal{B}^{\mathrm{DLIN}'}}(\lambda) \cdot \big(1 - \frac{1}{p}\big)^{-1}$. If $\mathbf{Adv}^{\mathrm{DLIN}}(\lambda)$ denotes the maximal advantage of any PPT distinguisher against the DLIN assumption in $\mathbb{G}$, the probability to have $F_{6.1} \wedge \mathsf{Good}$ can be bounded as $\Pr[F_{6.1} \wedge \mathsf{Good}] \leq \frac{3}{2} \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\lambda) \cdot \left(1 - \frac{1}{p}\right)^{-1}$ in $\mathsf{Game}_{6.1}$. This eventually yields the stated result

$$\Pr[F_6] \leq 7 \cdot \sqrt{(q_e + 1) \cdot (L + 1) \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\lambda) \cdot \left(1 - \frac{1}{p}\right)^{-1}}.$$

□

I More Efficient Adaptively Secure CCA2-Secure Threshold Cryptosystems from the DLIN and *k*-Linear Assumptions

As a use case for our relatively sound QA-NIZK proofs, we can construct a new robust non-interactive threshold encryption scheme based on the DLIN assumption and prove it secure against chosen-ciphertext attacks in the adaptive corruption setting [16, 28].

Threshold cryptosystems were initially suggested in [13, 22, 23]. In the static corruption setting, several non-interactive CCA-secure threshold systems have been described in the random oracle model [53, 27] and in the standard model [9, 14, 55].

Adaptively secure distributed cryptosystems with chosen-ciphertext security were proposed in [36, 1] but they require some interaction during the decryption process. Non-interactive solutions were put forth in [41, 42] but, as we will see, they are less efficient than the solution proposed here.

Consider the DLIN-based cryptosystem based on 1-universal hash proof systems where the ciphertext $(C_1, C_2, C_3, C_0) = (f^{\theta_1}, h^{\theta_2}, g^{\theta_1+\theta_2}, M \cdot X_1^{\theta_1} \cdot X_2^{\theta_2})$ is decrypted as $M = C_0 \cdot C_1^{-x_1} C_2^{-x_2} C_3^{-x_0}$, where $(X_1, X_2) = (f^{x_1} g^y, h^{x_2} g^{x_0})$ is the public key and (x_1, x_2, x_0) is the private key. In [42], chosen-ciphertext security was achieved using a *publicly* verifiable one-time simulation-sound proof of well-formedness for (C_1, C_2, C_3). In the security proof, the one-time simulation-soundness property guarantees that the adversary is unable to trick the decryption oracle into returning the decryption of an invalid ciphertext, by generating a fake proof for an invalid triple (C_1, C_2, C_3). For this reason, the specific private key (x_1, x_2, x_0) used by the reduction remains perfectly hidden. Consequently, if the challenge ciphertext is computed by choosing a random tuple $(C_1, C_2, C_3) \in_R \mathbb{G}^3$ and computing $C_0 = M \cdot C_1{}^{x_1} \cdot C_2{}^{x_2} \cdot C_3{}^{x_0}$, the plaintext M is independent of the adversary's view. To prove adaptive security in the threshold setting, [42] took advantage of the fact that the private key (more precisely, all private key shares) is known to the reduction at all times in the Cramer-Shoup paradigm.

A similar approach was taken in[6] [37], where a different method was used to achieve a form of one-time simulation-soundness. In combination with relatively sound proofs [37], the techniques of Jutla and Roy [38] reduce the size of ciphertexts to 9 group elements under the DLIN assumption.

Here, as already suggested in [43], we obtain shorter ciphertexts by using linearly homomorphic signatures. We include in the public key the verification key of a one-time linearly homomorphic SPS for $n = 3$ as well as signatures on $(f, 1_{\mathbb{G}}, g)$ and $(1_{\mathbb{G}}, h, g)$. This allows publicly deriving a homomorphic signature (z, r, u) on the vector $(C_1, C_2, C_3) = (f^{\theta_1}, h^{\theta_2}, g^{\theta_1+\theta_2})$ and each ciphertext consists of $(z, r, u, C_0, C_1, C_2, C_3)$. In the security proof, the signature (z, r, u) serves as evidence that (C_1, C_2, C_3) has the right form at each pre-challenge decryption query: in order to generate a proof for a false statement, the adversary has to break the security of the homomorphic signature, by deriving a signature on a vector (C_1, C_2, C_3) outside the span of $(f, 1, g)$ and $(1, h, g)$.

While this technique does provide IND-CCA1 security, the scheme remains malleable and thus vulnerable to post-challenge decryption queries. This is where the relatively sound proof system of Section 4 comes into play. By using (C_0, C_1, C_2, C_3) as a label in the relatively sound proof that (C_1, C_2, C_3) lives in $\mathrm{span}((f, 1, g), (1, h, g))$, we can make sure that, with all but negligible probability, the reduction will never accept a proof for a malformed (C_1, C_2, C_3) after the challenge phase without breaking the DLIN assumption. The key idea of the techniques of [37] is to guarantee that

[6] Although it was not mentioned in [37], relatively sound proofs can be used to acquire CCA2 security in the threshold setting as well, as will be emphasized later on.

the adversary will not be able to send a decryption query for which the private verifier and the public verifier disagree on (C_1, C_2, C_3).

I.1 Construction

In the threshold setting, the construction can be seen as a DLIN-based version of the Cramer-Shoup encryption scheme [20] (which is identical to the scheme in [52]), where the ciphertext components (C_1, C_2, C_3) and the designated verifier proof C_4 are additionally signed using a homomorphic signature. The scheme goes as follows.

Keygen(λ, t, N)**:** choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ and do the following.

1. Choose $g, f, h \overset{R}{\leftarrow} \mathbb{G}$.
2. Choose random $x_0, x_1, x_2 \overset{R}{\leftarrow} \mathbb{Z}_p$, $y_0, y_1, y_2 \overset{R}{\leftarrow} \mathbb{Z}_p$ and $w_0, w_1, w_2 \overset{R}{\leftarrow} \mathbb{Z}_p$ in order to compute $X_1 = f^{x_1} g^{x_0}$, $X_2 = h^{x_2} g^{x_0}$, $Y_1 = f^{y_1} g^{y_0}$, $Y_2 = h^{y_2} g^{y_0}$ and $W_1 = f^{w_1} g^{w_0}$, $W_2 = h^{w_2} g^{w_0}$.
3. Generate a Groth-Sahai CRS $(\boldsymbol{f_1}, \boldsymbol{f_2}, \boldsymbol{f_3})$ to be used for proving the validity of decryption shares. Namely, choose $f_1, f_2 \overset{R}{\leftarrow} \mathbb{G}$ as well as $\phi_1, \phi_2 \overset{R}{\leftarrow} \mathbb{Z}_p$ and define vectors
$$\boldsymbol{f}_1 = (f_1, 1, g), \qquad \boldsymbol{f}_2 = (1, f_2, g) \qquad \boldsymbol{f}_3 = \boldsymbol{f}_1^{\phi_1} \cdot \boldsymbol{f}_2^{\phi_2} \cdot (1, 1, g).$$
4. Choose random polynomials $P_1[Z], P_2[Z], P_0[Z] \in \mathbb{Z}_p[Z]$ of degree $t-1$ such that $P_1(0) = x_1$, $P_2(0) = x_2$ and $P_0(0) = x_0$. Set $X_{i,1} = f^{P_1(i)} g^{P_0(i)}$ and $X_{i,2} = h^{P_2(i)} g^{P_0(i)}$ for $i = 1$ to N.
5. Choose a collision-resistant hash function $H : \{0,1\}^* \to \mathbb{Z}_p$.
6. Generate a key pair for the one-time linearly homomorphic signature of Section 2.5 with $n = 7$. Let $\big(g_z, g_r, h_z, h_u, \{(g_i, h_i)\}_{i=1}^7\big)$ be the public key and let $\{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^7$ be the corresponding private key.
7. Generate one-time homomorphic signatures $\{(Z_j, R_j, U_j)\}_{j=1}^4$ on the independent vectors
$$\boldsymbol{h}_1 = (f, 1, g, Y_1, 1, 1, 1) \in \mathbb{G}^7, \qquad \boldsymbol{h}_2 = (1, h, g, Y_2, 1, 1, 1) \in \mathbb{G}^7$$
$$\boldsymbol{h}_3 = (1, 1, 1, W_1, f, 1, g) \in \mathbb{G}^7, \qquad \boldsymbol{h}_4 = (1, 1, 1, W_2, 1, h, g) \in \mathbb{G}^7.$$
8. Define decryption key shares $\mathbf{SK} = (SK_1, \ldots, SK_N)$ as $SK_i = (P_1(i), P_2(i), P_0(i)) \in \mathbb{Z}_p^3$ for each $i \in \{1, \ldots, N\}$. The vector $\mathbf{VK} = (VK_1, \ldots, VK_N)$ of verification keys is defined as $VK_i = (X_{i,1}, X_{i,2}) \in \mathbb{G}^2$ for each $i \in \{1, \ldots, N\}$. The public key is defined to be
$$PK = \Big(g,\ \boldsymbol{f}_1,\ \boldsymbol{f}_2,\ \boldsymbol{f}_3,\ X_1,\ X_2,\ Y_1,\ Y_2,\ W_1,\ W_2,\ g_z,\ g_r,\ h_z,\ h_u,$$
$$\{g_i, h_i\}_{i=1}^7,\ \{(Z_j, R_j, U_j)\}_{j=1}^4,\ H \Big).$$

Encrypt(M, PK)**:** to encrypt a message $M \in \mathbb{G}$, conduct the following steps.

1. Choose $\theta_1, \theta_2 \overset{R}{\leftarrow} \mathbb{Z}_p$ and compute
$$C_0 = M \cdot X_1^{\theta_1} \cdot X_2^{\theta_2}, \qquad C_1 = f^{\theta_1}, \qquad C_2 = h^{\theta_2}, \qquad C_3 = g^{\theta_1 + \theta_2} \qquad C_4 = (W_1^\alpha Y_1)^{\theta_1} \cdot (W_2^\alpha Y_2)^{\theta_2},$$
where $\alpha = H(C_0, C_1, C_2, C_3) \in \mathbb{Z}_p$.
2. Construct a linearly homomorphic signature (Z, R, U) on $(C_1, C_2, C_3, C_4, C_1^\alpha, C_2^\alpha, C_3^\alpha) \in \mathbb{G}^7$. Namely, compute
$$Z = Z_1^{\theta_1} \cdot Z_2^{\theta_2} \cdot Z_3^{\theta_1 \cdot \alpha} \cdot Z_4^{\theta_2 \cdot \alpha}, \qquad R = R_1^{\theta_1} \cdot R_2^{\theta_2} \cdot R_3^{\theta_1 \cdot \alpha} \cdot R_4^{\theta_2 \cdot \alpha}, \qquad U = U_1^{\theta_1} \cdot U_2^{\theta_2} \cdot U_3^{\theta_1 \cdot \alpha} \cdot U_4^{\theta_2 \cdot \alpha}$$
3. Output the ciphertext
$$C = (C_0, C_1, C_2, C_3, C_4, Z, R, U) \in \mathbb{G}^8 \qquad (21)$$

Ciphertext-Verify(PK, C)**:** parse C as per (21). Compute $\alpha = H(C_0, C_1, C_2, C_3) \in \mathbb{Z}_p$ and return 1 if and only if

$$1_{\mathbb{G}_T} = e(g_z, Z) \cdot e(g_r, R) \cdot \prod_{i=1}^{3} e(g_i \cdot g_{i+4}^{\alpha}, C_i) \cdot e(g_4, C_4)$$
$$1_{\mathbb{G}_T} = e(h_z, Z) \cdot e(h_u, U) \cdot \prod_{i=1}^{3} e(h_i \cdot h_{i+4}^{\alpha}, C_i) \cdot \cdot e(h_4, C_4),$$

Share-Decrypt(PK, i, SK_i, C)**:** on inputs $SK_i = (P_1(i), P_2(i), P_0(i)) \in \mathbb{Z}_p^3$ and C, return $(i, \perp)$ in the event that **Ciphertext-Verify**$(PK, C) = 0$. Otherwise, compute $\hat{\mu}_i = (\nu_i, \boldsymbol{C}_{P_1}, \boldsymbol{C}_{P_2}, \boldsymbol{C}_{P}, \pi_{\mu_i})$ which consists of a partial decryption $\nu_i = C_1^{P_1(i)} \cdot C_2^{P_2(i)} \cdot C_3^{P_0(i)}$, commitments $\boldsymbol{C}_{P_1}, \boldsymbol{C}_{P_2}, \boldsymbol{C}_{P_0}$ to exponents $P_1(i), P_2(i), P_0(i) \in \mathbb{Z}_p$ and a proof π_{ν_i} that these satisfy the equations

$$\nu_i = C_1^{P_1(i)} \cdot C_2^{P_2(i)} \cdot C_3^{P_0(i)}, \qquad X_{i,1} = f^{P_1(i)} g^{P_0(i)}, \qquad X_{i,2} = h^{P_2(i)} g^{P_0(i)}. \tag{22}$$

The commitments $\boldsymbol{C}_{P_1}, \boldsymbol{C}_{P_2}, \boldsymbol{C}_{P_0}$ and the proof π_{ν_i} are generated using the CRS $\mathbf{f} = (\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3)$.

Share-Verify$(PK, VK_i, C, (i, \hat{\mu}_i))$**:** parse the ciphertext C as $(C_0, C_1, C_2, C_3, C_4, Z, R, U)$ and VK_i as $(X_{i,1}, X_{i,2}) \in \mathbb{G}^2$. If $\hat{\mu}_i = \perp$ or $\hat{\mu}_i$ cannot be properly parsed as $(\nu_i, \boldsymbol{C}_{P_1}, \boldsymbol{C}_{P_2}, \boldsymbol{C}_{P_0}, \pi_{\mu_i})$, return 0. Otherwise, return 1 if π_{μ_i} is a valid proof. In any other situation, return 0.

Combine$(PK, \mathbf{VK}, C, \{(i, \hat{\mu}_i)\}_{i \in S})$**:** for each $i \in S$, parse the share $\hat{\mu}_i$ as $(\nu_i, \boldsymbol{C}_{P_1}, \boldsymbol{C}_{P_2}, \boldsymbol{C}_{P}, \pi_{\mu_i})$ and return $\perp$ if **Share-Verify**$(PK, C, (i, \hat{\mu}_i)) = 0$. Otherwise, compute $\nu = \prod_{i \in S} \nu_i^{\Delta_{i,S}(0)}$, which equals $\nu = C_1^{x_1} \cdot C_2^{x_2} \cdot C_3^{x_0} = X_1^{\theta_1} \cdot X_2^{\theta_2}$ and in turn reveals $M = C_0/\nu$.

If each element has a 256-bit representation on BN curves [4] at the 128-bit security level, the ciphertext overhead amounts to 1792 bits. The DLIN-based scheme of [42] has a ciphertext overhead comprised of 14 group elements and a one-time signature with its verification key (or 4864 bits using Groth's one-time signature [32]). The results of Escala *et al.* [25] reduce this overhead to 3328 bits. The recent techniques of Jutla and Roy [37, 38] – which also work in the threshold setting although it was not explicitly stated in [37] – lead to ciphertexts comprised of 9 group elements under the DLIN assumption and $3k + 3$ under the k-linear assumption. Under DLIN, we thus further compress ciphertexts by 11% while relying on the same assumption and retaining tight security[7].

Under the k-linear assumption, our improvement becomes more important as the ciphertext reduces to $2k + 4$ group elements. Specifically, we need $k + 1$ elements for the homomorphic signature of Appendix D, another $k + 1$ elements to contain the k-linear instance, one element for the Cramer-Shoup-like proof π_0 and one element to carry the plaintext. This allows saving $k - 1$ group elements with respect to the techniques of [37, 38].

We believe this result to be of importance as these schemes can potentially serve as building blocks for protocols in the multi-linear setting [29, 19]. Indeed, the $(k - 1)$-linear problem is easy in groups equipped with a k-linear map (as shown in, *e.g.*, [25]) but we can hope for instantiations where the k-linear assumption holds, as seems to be the case in [19].

From a computational standpoint, the validity of a ciphertext only requires to compute a product of 7 pairings. Under the the DLIN assumption, the framework of [42] requires a product of 12 pairings in the ciphertext verification algorithm.

[7] Note that the techniques of Lewko [40] can be applied to the scheme of [41] to get a DLIN-based system where ciphertexts contain 7 group elements and a one-time key pair (SVK, σ). However, the reduction involves a degradation factor proportional to the number of decryption queries.

I.2 Security

We prove security in the sense of a definition which is identical to Definition 1 with the difference that there is no evaluation key SK_h, no evaluation oracle and no RevHK oracle.

As in the scheme of [37], the security proof appeals to the private verification algorithm while the scheme itself only uses the public verification algorithm.

While it would be possible to rely on the relative zero-knowledge and relative soundness properties of the proof system in a modular way, we obtain a better exact security via a direct analysis.

Theorem 4. *The above threshold cryptosystem is IND-CCA secure against adaptive corruptions assuming that: (i) H is collision-resistant; (ii) The DLIN assumption holds in $\mathbb{G}$. More precisely, the advantage of any PPT adversary $\mathcal{A}$ is at most*

$$\mathbf{Adv}(\mathcal{A}) \leq \mathbf{Adv}^{\text{CR-hash}}(\lambda) + 3 \cdot \mathbf{Adv}^{\text{DLIN}}(\lambda) + \frac{2q+1}{2^{\lambda} - q}, \tag{23}$$

where q is the number of decryption queries and the first term of the right-hand-side member accounts for the maximal advantage of any PPT collision-finding algorithm for H.

Proof. The proof uses of a sequence of games starting with the real attack game and ending with a game where the adversary $\mathcal{A}$ has no advantage. For each i, S_i stands for the event that the challenger $\mathcal{B}$ outputs 1 at the end of Game_i.

Game_1**:** is the real attack game. In details, the adversary is given the public key PK and the set of verification keys $\mathbf{VK} = (VK_1, \ldots, VK_N)$. At each corruption query $i \in \{1, \ldots, n\}$, the challenger $\mathcal{B}$ reveals the queried private key share $SK_i = (P_1(i), P_2(i), P_0(i))$. At each decryption query, $\mathcal{B}$ faithfully runs the real shared decryption algorithm. In the challenge phase, the adversary $\mathcal{A}$ chooses messages $M_0, M_1 \in \mathbb{G}$ and obtains $C^\star = (C_0^\star, C_1^\star, C_2^\star, C_3^\star, C_4^\star, Z^\star, R^\star, U^\star)$ which is an encryption of M_β, for some random coin $\beta \stackrel{R}{\leftarrow} \{0,1\}$ flipped by $\mathcal{B}$. We can assume w.l.o.g. that $(C_1^\star, C_2^\star, C_3^\star)$ are computed at the beginning of the game as they do not depend on M_β.

In the second phase, $\mathcal{A}$ makes more adaptive queries under the restriction of not asking for a partial decryption of $C^\star$ or for more than $t-1$ private key shares throughout the entire game. Eventually, $\mathcal{A}$ halts and outputs β'. At this point, $\mathcal{B}$ outputs 1 if $\beta = \beta'$ and 0 otherwise.

Game_2**:** This game is like Game_1 except that the challenger $\mathcal{B}$ halts and outputs a random bit in the event that, before the challenger phase, $\mathcal{A}$ queries the partial decryption oracle on a ciphertext $C = (C_0, C_1, C_2, C_3, C_4, Z, R, U)$ such that $(C_1, C_2, C_3) = (C_1^\star, C_2^\star, C_3^\star)$. Since $(C_1^\star, C_2^\star, C_3^\star)$ are invisible to $\mathcal{A}$ until the challenge phase, this event can only occur with probability q/p, so that $|\Pr[S_2] - \Pr[S_1]| < q/p$.

Game_3**:** We introduce another failure event F_3 and let $\mathcal{B}$ halt and output a random bit if this event occurs. We define F_3 as the event that $\mathcal{A}$ makes a decryption query involving a valid ciphertext $C = (C_0, C_1, C_2, C_3, C_4, Z, R, U)$ such that $H(C_0, C_1, C_2, C_3) = H(C_0^\star, C_1^\star, C_2^\star, C_3^\star)$ but $(C_0^\star, C_1^\star, C_2^\star, C_3^\star) \neq (C_0, C_1, C_2, C_3)$.

We see that Game_3 and Game_2 are identical until event F_3 occurs, which would contradict the collision-resistance of H. We thus have $|\Pr[S_3] - \Pr[S_2]| \leq \Pr[F_3] \leq \mathbf{Adv}^{\text{CR-hash}}(\lambda)$. In subsequent games, if we define the values $\alpha = H(C_0, C_1, C_2, C_3)$ and $\alpha^\star = H(C_0^\star, C_1^\star, C_2^\star, C_3^\star)$, we will henceforth assume that $\alpha \neq \alpha^\star$ for each decryption query $C = (C_0, C_1, C_2, C_3, C_4, Z, R, U)$.

Game_4**:** In this game, we modify the decryption oracle and reject all post-challenge decryption queries $(C_0, C_1, C_2, C_3, C_4, Z, R, U)$ such that $(C_0, C_1, C_2, C_3, C_4) = (C_0^\star, C_1^\star, C_2^\star, C_3^\star, C_4^\star)$. Clearly Game_4 is identical to Game_3 until $\mathcal{B}$ rejects a ciphertext that would not have been rejected in Game_3.

If we call the latter event F_4, we find that $|\Pr[S_4] - \Pr[S_3]| \leq \Pr[F_4]$. Since F_4 necessarily implies $(Z, R, U) \neq (Z^\star, R^\star, U^\star)$, any occurrence of F_4 necessarily provides $\mathcal{A}$ with two distinct signatures on the same vector $(C_1^\star, C_2^\star, C_3^\star, C_4^\star, C_1^{\star\alpha^\star}, C_2^{\star\alpha^\star}, C_3^{\star\alpha^\star})$, which in turn breaks the SDP assumption by the specific property of the linearly homomorphic signature (see Section 2.5). It comes that $\Pr[F_4] \leq \mathbf{Adv}^{\mathrm{SDP}}(\mathcal{B})$.

Game_5: We modify the generation of $C^\star = (C_0^\star, C_1^\star, C_2^\star, C_3^\star, C_4^\star, Z^\star, R^\star, U^\star)$ in the challenge phase. Specifically, instead of computing $C_0^\star = M_\beta \cdot X_1^{\theta_1} \cdot X_2^{\theta_2}$ and $C_4^\star = (W_1^{\alpha^\star} Y_1)^{\theta_1} (W_2^{\alpha^\star} Y_2)^{\theta_2}$, where $\theta_1 = \log_f(C_1^\star)$ and $\theta_2 = \log_h(C_2^\star)$, the challenger $\mathcal{B}$ now computes $C_0^\star = M_\beta \cdot C_1^{\star x_1} \cdot C_2^{\star x_2} \cdot C_3^{\star x_0}$ and $C_4^\star = C_1^{\star y_1 + \alpha^\star w_1} \cdot C_2^{\star y_2 + \alpha^\star w_2} \cdot C_3^{\star y_0 + \alpha^\star w_0}$, with $\alpha^\star = H(C_0^\star, C_1^\star, C_2^\star, C_3^\star)$. Likewise, instead of using the encryption exponents (θ_1, θ_2) to derive a one-time linearly homomorphic signature $(Z^\star, R^\star, U^\star)$ from the public key, the challenger $\mathcal{B}$ uses $\{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^7$ and computes

$$Z^\star = \prod_{i=1}^{7} C_i^{\star -\chi_i} \qquad R^\star = \prod_{i=1}^{7} C_i^{\star -\gamma_i} \qquad U^\star = \prod_{i=1}^{7} C_i^{\star -\delta_i}, \tag{24}$$

where $(C_5^\star, C_6^\star, C_7^\star) = (C_1^{\star\alpha}, C_2^{\star\alpha}, C_3^{\star\alpha})$.

This change is only conceptual since $C_0^\star$ still equals $C_0^\star = M_\beta \cdot X_1^{\theta_1} \cdot X_2^{\theta_2}$ and the distribution of $(C_4^\star, Z^\star, R^\star, U^\star)$ has not changed either. We thus have $\Pr[S_5] = \Pr[S_4]$.

Game_6: Here, we modify the decryption oracle and make use of the exponents $(y_0, y_1, y_2, w_0, w_1, w_2)$ that were chosen by $\mathcal{B}$ during the key generation phase. Namely, the challenger $\mathcal{B}$ does not only reject all ciphertexts $(C_0, C_1, C_2, C_3, C_4, Z, R, U)$ such that (Z, R, U) does not form a valid signature on $(C_1, C_2, C_3, C_4, C_1^\alpha, C_2^\alpha, C_3^\alpha)$ but also rejects those for which

$$C_4 \neq C_1^{y_1 + \alpha \cdot w_1} \cdot C_2^{y_2 + \alpha \cdot w_2} \cdot C_3^{y_0 + \alpha \cdot w_0},$$

where $\alpha = H(C_0, C_1, C_2, C_3)$. We raise a failure event F_6, which causes $\mathcal{B}$ to halt and output a random bit if it occurs. This event F_6 is defined to be the event that the adversary $\mathcal{A}$ queries the decryption oracle on a ciphertext that gets rejected in Game_6 and would not have been rejected in Game_5. Since Game_6 is identical to Game_5 until F_6 occurs, we have

$$\Pr[S_6] = \Pr[S_6 \wedge \neg F_6] + \frac{1}{2} \cdot \Pr[F_6] = \Pr[S_5] + \frac{1}{2} \cdot \Pr[F_6].$$

At the same time, Lemma 3 shows that $\Pr[F_6] \leq \mathbf{Adv}^{\mathrm{SDP}}(\lambda) + \frac{1}{p}$. We remark that a side-effect of this modified decryption oracle is that it now rejects all post-challenge decryption queries $(C_0, C_1, C_2, C_3, C_4, Z, R, U)$ such that $(C_0, C_1, C_2, C_3) = (C_0^\star, C_1^\star, C_2^\star, C_3^\star)$ but $C_4 \neq C_4^\star$.

Since F_6 is always efficiently detectable by the challenger $\mathcal{B}$, we can build an efficient DLIN distinguisher if the probability of event F_6 increases when $C_3^\star$ is tampered with in the challenge ciphertext as in the next game.

Game_7: This game is identical to Game_6 with one modification in the challenge ciphertext. Instead of setting $C_3^\star = g^{\theta_1 + \theta_2}$, where $\theta_1 = \log_f(C_1^\star)$ and $\theta_2 = \log_h(C_2^\star)$, we choose it as $C_3^\star \overset{R}{\leftarrow} \mathbb{G}$. The linearly homomorphic signature $(Z^\star, R^\star, U^\star)$ is computed according to (24), as previously. Under the DLIN assumption in $\mathbb{G}$, this modification should have no noticeable impact on $\mathcal{A}$'s behavior. In particular, we have $|\Pr[S_7] - \Pr[S_6]| \leq \mathbf{Adv}^{\mathrm{DLIN}}(\lambda)$.

Game_8: We modify the partial decryption oracle and replace the non-interactive proofs contained in decryption shares $\hat{\mu_i}$ by simulated NIZK proofs. This entails to turn $(\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3)$ into a perfectly hiding Groth-Sahai CRS (where $\boldsymbol{f_3}$ is in span$(\boldsymbol{f_1}, \boldsymbol{f_2})$) and non-interactive proofs for multi-exponentiation equations are simulated using the trapdoor of the simulated CRS. Under the DLIN assumption, this change is not noticeable by $\mathcal{A}$ and we have $|\Pr[S_8] - \Pr[S_7]| \leq \mathbf{Adv}^{\mathrm{DLIN}}(\lambda)$.

Game_9: In this game, we modify again the decryption oracle and make use of the discrete logarithms $\alpha_f = \log_g(f)$ and $\alpha_h = \log_g(h)$. Since we are done with the transition consisting in replacing $C_3^\star$ by a random element, we are free to use (α_f, α_h) from this point forward. We thus introduce a modification in the treatment of decryption queries $C = (C_0, C_1, C_2, C_3, C_4, Z, R, U)$. This, $\mathcal{B}$ rejects *all* ciphertexts C such that $C_3 \neq C_1^{1/\alpha_f} \cdot C_2^{1/\alpha_h}$. Otherwise, it answers as in Game_8.

If we define F_9 to be the event that $\mathcal{B}$ rejects a ciphertext which would not have been rejected in Game_8, we see that Game_9 and Game_8 are identical from $\mathcal{A}$'s view until F_9 occurs. Therefore it comes that

$$\Pr[S_9] \leq \Pr[S_9 \wedge \neg F_9] + \Pr[F_9] = \Pr[S_8] + \Pr[F_9].$$

The same arguments as in the proof of Cramer and Shoup show that $\Pr[F_9] \leq q/(p-q)$. More precisely, after i decryption queries, the adversary is left with $p - i$ equally likely candidates for the value of C_4 that would have been accepted by the private ciphertext validation algorithm. The probability that the i-th decryption query satisfies the test given that the first $i-1$ queries have failed it is thus at most $i/(p-i)$.

In Game_9, it is easy to see that $\mathcal{A}$ has no advantage whatsoever and we have $\Pr[S_9] = 1/2$. Indeed, in the challenge phase, we have $(C_1^\star, C_2^\star, C_3^\star) = (f^{\theta_1}, h^{\theta_2}, g^{\theta_1+\theta_2+\theta_3})$, with $\theta_1, \theta_2, \theta_3 \in_R \mathbb{Z}_p$, so that $C_0^\star$ can be written as $C_0^\star = M_\beta \cdot X_1^{\theta_1} \cdot X_2^{\theta_2} \cdot g^{\theta_3 \cdot x_0}$. The latter equality implies that, as long as $x_0 \in \mathbb{Z}_p$ is independent of $\mathcal{A}$'s view, so is the bit $\beta \in \{0, 1\}$.

We also note that, in Game_9, decryption shares $\hat{\mu_i}$ contain NIZK proofs that are simulated without using private key shares and thus leak no information about these. It comes that, as long as $\mathcal{A}$ does not corrupt more than $t-1$ servers, the only possible way to infer information about $x_0 = P(0)$ is to make decryption queries on invalid ciphertexts (*i.e.*, for which (C_1, C_2, C_3) lies outside the span of $\boldsymbol{f}$ and $\boldsymbol{h}$).

We thus find

$$|\Pr[S_1] - \frac{1}{2}| < \mathbf{Adv}^{\text{CR-hash}}(\lambda) + 2 \cdot \mathbf{Adv}^{\text{DLIN}}(\lambda) + 2 \cdot \mathbf{Adv}^{\text{SDP}}(\lambda) + \frac{2q+1}{p-q}.$$

Since any algorithm solving SDP immediately provides a DLIN distinguisher, we also have the inequality $\mathbf{Adv}^{\text{SDP}}(\lambda) \leq \frac{1}{2}\mathbf{Adv}^{\text{DLIN}}(\lambda)$, which yields

$$|\Pr[S_1] - \frac{1}{2}| < \mathbf{Adv}^{\text{CR-hash}}(\lambda) + 3 \cdot \mathbf{Adv}^{\text{DLIN}}(\lambda) + \frac{2q+1}{2^\lambda - q} \tag{25}$$

and the claimed result follows. □

Lemma 3. *In* Game_6*, the probability of event* F_6 *is at most* $\Pr[F_6] \leq \mathbf{Adv}^{\text{SDP}}(\lambda) + \frac{1}{p}$.

Proof. We show that, if event F_6 occurs with non-negligible probability ε in Game_6, there exists an efficient algorithm $\mathcal{B}$ that solves a SDP instance (g_z, g_r, h_z, h_u) with about the same probability. To this end, we first remark that F_6 can only occur for a decryption query $(C_0, C_1, C_2, C_3, C_4, Z, R, U)$ such that $(C_1, C_2, C_3, C_4, C_1^\alpha, C_2^\alpha, C_3^\alpha)$ is outside span$(\boldsymbol{h_1}, \boldsymbol{h_2}, \boldsymbol{h_3}, \boldsymbol{h_4})$. Indeed, otherwise, there exist integers $\theta_1, \theta_2 \in \mathbb{Z}_p$ such that $(C_1, C_2, C_3, C_4) = \big(f^{\theta_1}, h^{\theta_2}, g^{\theta_1+\theta_2}, (W_1^\alpha Y_1)^{\theta_1} (W_2^\alpha Y_2)^{\theta_2}\big)$, in which case we always have $C_4 = C_1^{y_1+\alpha w_1} \cdot C_2^{y_2+\alpha w_2} \cdot C_3^{y_0+\alpha w_0}$ and the rejection rule of Game_6 does not apply.

Using the technique of [43][Theorem 1], we show that event F_6 implies an algorithm solving the given SDP instance with nearly the same probability. Algorithm $\mathcal{B}$ begins by setting up $g_i = {g_z}^{\chi_i} {g_r}^{\gamma_i}$ and $h_i = {h_z}^{\chi_i} {h_u}^{\delta_i}$, with $\chi_i, \gamma_i, \delta_i \xleftarrow{R} \mathbb{Z}_p$ for $i \in \{1, \ldots, 7\}$. Other public key components are generated as in the real scheme and the public key is given to the adversary.

Throughout the game, the reduction $\mathcal{B}$ answers $\mathcal{A}$'s decryption queries in the same way as in Game_6. By hypothesis, $\mathcal{A}$ must query the decryption of a ciphertext $(C_0, C_1, C_2, C_3, C_4, Z, R, U)$ such that (Z, R, U) is a valid linearly homomorphic signature on the vector $(C_1, C_2, C_3, C_4, C_1^\alpha, C_2^\alpha, C_3^\alpha)$, where $\alpha = H(C_0, C_1, C_2, C_3)$, but $C_4 \neq C_1^{y_1+\alpha w_1} \cdot C_2^{y_2+\alpha w_2} \cdot C_3^{y_0+\alpha w_0}$, which implies that the vector is not in span$(\boldsymbol{h_1}, \boldsymbol{h_2}, \boldsymbol{h_3}, \boldsymbol{h_4})$. When $\mathcal{B}$ detects this event, it defines $(C_5, C_6, C_7) = ({C_1}^{\alpha}, {C_2}^{\alpha}, {C_3}^{\alpha})$ and computes its own signature

$$(Z^{\dagger}, R^{\dagger}, U^{\dagger}) = (\prod_{i=1}^{7} {C_i}^{-\chi_i}, \prod_{i=1}^{7} {C_i}^{-\gamma_i}, \prod_{i=1}^{7} {C_i}^{-\delta_i}) \tag{26}$$

on $(C_1, C_2, C_3, C_4, C_5, C_6, C_7)$. We claim that, with overwhelming probability,

$$(Z^{\ddagger}, R^{\ddagger}, U^{\ddagger}) = \Big(\frac{Z}{Z^{\dagger}}, \frac{R}{R^{\dagger}}, \frac{U}{U^{\dagger}}\Big)$$

is a non-trivial solution to the SDP instance since $Z^{\ddagger} \neq 1_{\mathbb{G}}$ with all but negligible probability.

To see this, we first note that the vector $(\chi_1, \ldots, \chi_7)$ is independent of $\mathcal{A}$'s view before the challenge phase. Hence, since $(C_1, C_2, C_3, C_4, C_5, C_6, C_7)$ is linearly independent of $(\boldsymbol{h}_1, \boldsymbol{h}_2, \boldsymbol{h}_3, \boldsymbol{h}_4)$, the adversary $\mathcal{A}$ can only predict $Z^{\dagger}$ (as it is computed in (26)) with negligible probability $1/p$. The probability $\Pr[F_6]$ can thus be bounded as $\Pr[F_6] \leq \mathbf{Adv}_{\mathcal{B}}^{\mathrm{SDP}}(\lambda) + \frac{1}{p}$. □

In the proof of the above theorem, the relative simulation-soundness property of the proof system is notably used in the transition from Game_8 to Game_9. In order to obtain a tighter reduction, we chose not to rely on this property in a modular way. In the modular approach, we would have to build an algorithm $\mathcal{B}^{rs}$ that contradicts this property using an adversary for which event F_9 occurs with non-negligible probability. This algorithm $\mathcal{B}^{rs}$ would have to interact with the relative soundness challenger for a *given* language $\rho \in \mathbb{G}^{2\times 3}$ for which $\mathcal{B}^{rs}$ does *not* have the underlying matrix $\mathbf{A} \in \mathbb{Z}_p^{2\times 3}$ of discrete logarithms. For this reason, $\mathcal{B}^{rs}$ would not be able to efficiently detect when F_9 occurs. To break the relative soundness property, $\mathcal{B}^{rs}$ would have to guess the decryption query for which this event occurs, which is only possible with probability $1/q$. In the exact security result (23), we would thus lose a multiplicative factor of $O(q)$.