Efficient Signcryption with Key Privacy from Gap Diffie-Hellman Groups

Benoît Libert Jean-Jacques Quisquater

UCL Crypto Group
Place du Levant, 3. B-1348 Louvain-La-Neuve. Belgium
{libert,jjq}@dice.ucl.ac.be –– http://www.uclcrypto.org/

Abstract. This paper proposes a new public key authenticated encryption (signcryption) scheme based on the Diffie-Hellman problem in Gap Diffie-Hellman groups. This scheme is built on the scheme proposed by Boneh, Lynn and Shacham in 2001 to produce short signatures. The idea is to introduce some randomness into this signature to increase its level of security in the random oracle model and to re-use that randomness to perform encryption. This results in a signcryption protocol that is more efficient than any combination of that signature with an El Gamal like encryption scheme. The new scheme is also shown to satisfy really strong security notions and its strong unforgeability is tightly related to the Diffie-Hellman assumption in Gap Diffie-Hellman groups.

Keywords. signcryption, pairings, Gap Diffie-Hellman groups, provable security

1 Introduction

The concept of public key signcryption schemes was proposed by Zheng in 1997 [38]. The purpose of this kind of primitive is to perform encryption and signature in a single logical step in order to obtain confidentiality, integrity, authentication and non-repudiation more efficiently than the sign-then-encrypt approach. The drawback of this latter solution is to expand the final ciphertext size (this could be impractical for low bandwidth networks) and increase the sender and receiver's computing time. Several efficient signcryption schemes have been proposed since 1997. The original scheme proposed in [38] was based on the discrete logarithm problem but no formal security proofs were given. Zheng's original construction was only proven secure in 2002 [3] by Baek et al. who described a formal security model in a multi-user setting. In 2000, Steinfeld and Zheng [33] proposed another scheme for which the unforgeability of ciphertexts relies on the intractability of the factoring problem but they provided no proof of chosen ciphertext security.

The drawback of the previously cited solutions is that they do not offer easy non-repudiation of ciphertexts: a recipient cannot prove to a third party that some plaintext was actually signcrypted by the sender. Bao and Deng [5] proposed a method to add universal verifiability to Zheng's cryptosystem but their scheme was shown [31] to leak some information about the plaintext as other schemes described in [37] and [23]. The latter schemes can easily be modified to fix their problem but no strong guarantee of unforgeability can be obtained currently. In the discrete logarithm setting, another scheme was shown in [31] to be chosen ciphertext secure under the Gap Diffie-Hellman assumption. It was a signcryption scheme built on a modified version of the DSA signature scheme which is not provably secure currently. As a consequence, no proof of unforgeability could be found for that scheme. An RSA-based scheme was described by Malone-Lee and Mao [24] who provided proofs for both unforgeability under chosen-message attacks and chosen ciphertext security. Unfortunately, they only considered a security in a single-user setting rather than the more realistic multi-user setting. Furthermore, the security of that scheme is only loosely related to the RSA assumption. However, none of these schemes is provably secure against insider attacks: in some of them, an attacker learning some user's private key can recover all messages previously signcrypted by that user while some others are only proven secure against outsider attacks.

In 2002, An et al. [1] presented an approach consisting in performing signature and encryption in parallel: a plaintext is first transformed into a pair (c, d) made of a commitment c and a de-commitment d in such a way that c reveals no information about m while the pair (c, d) allows recovering m. Once he completed the transformation, the signer can jointly sign c and encrypt d in parallel using appropriate encryption and signature schemes. The de-signcryption operation is then achieved by the recipient in a parallel fashion: the signature on c is verified while d is decrypted and the pair (c, d) is then used to recover the plaintext. This method decreases the computation time to signcrypt a message to the maximum of the times required by the underlying encryption and signature processes. This parallel approach was further investigated by Pieprzyk and Pointcheval [27] who proposed to use a $(2, 2)$-Shamir secret sharing as a commitment scheme: a plaintext is first splitted into two shares s_1, s_2 which do not individually reveal any information on m. s_1 is used as a commitment and signed while s_2 is encrypted as a de-commitment. The authors of [27] also gave a construction allowing them to integrate any one-way encryption system (such as the basic RSA) with a weakly secure signature (non-universally forgeable signatures in fact) into a chosen ciphertext secure and existentially unforgeable signcryption scheme.

Dodis et al. [15] proposed another technique to perform parallel signcryption. Their method consists in a Feistel probabilistic two-paddings which can be viewed as a generalization of other existing probabilistic paddings (OAEP, OAEP+, PSS-R,etc.) and involve a particular kind of commitment schemes. The authors of [15] showed that their construction also allows optimal exact security, flexible key management, compatibility with PKCS standards and has other interesting properties. On the other hand, all parallel signcryption propositions require the recipient of a message to know from whom a ciphertext emanates before beginning to verify the signature in parallel with the decryption operation. A trivial solution to this problem is to append a tag containing the sender's identity to the ciphertext but this is not compatible with the notion of ciphertext anonymity formalized by Boyen in [12] (intuitively, this notion expresses the inability for someone observing a ciphertext to determine who the sender is nor to whom it is intended) that can be a desirable feature in many applications (see [12] for examples).

With the motivation of achieving truly anonymous and authenticated communications, this paper proposes a new Diffie-Hellman-based signcryption scheme which satisfies strong security notions: chosen-ciphertext security against insider attacks (except the hybrid composition proposed in [21] and the identity-based scheme described in [12], no discrete logarithm based authenticated encryption method was formally proven secure in such a model before), strong unforgeability against chosen-message attacks, ciphertext anonymity in the sense of [12] (this is an extension of the notion of key privacy proposed in [7] to the signcryption case). We also prove that it satisfies a new security notion that is related to the one of ciphertext anonymity and that we call 'key invisibility'. We show that the scheme's strong unforgeability is tightly related to the hardness of the Diffie-Hellman problem unlike the scheme proposed in [12] whose proof of unforgeability relies on Pointcheval and Stern's forking lemma and thus only provides a loose reduction to a computational problem. In fact, except the hybrid construction of [21] (whose semantic security is based on the stronger hash oracle Diffie-Hellman assumption) our scheme appears to be the first discrete-logarithm-based signcryption protocol whose (strong) unforgeabililty is proven to be tightly related to the Diffie-Hellman problem. Unlike [1, 15, 27], our protocol is sequential but it is efficient and does not require the recipient of a message to know who is the sender before starting the de-signcryption process. Our scheme draws inspiration from a construction due to Boyen [12] and makes use of the properties of bilinear maps over Gap Diffie-Hellman groups (in fact, the structure of these groups is extensively exploited in the security proofs). Before describing our scheme, we first recall the properties of these maps in section 2. The section 3 formally describes the security notions that our scheme, depicted in section 4, is shown to satisfy in the security analysis presented in section 5.

2 Preliminaries

2.1 Overview of pairings

Definition 1. *Let k be a security parameter and q be a k-bit prime number. Let us consider groups $(\mathbb{G}_1, +)$, $(\mathbb{G}_2, +)$ and $(\mathbb{G}_T, \cdot)$ of the same prime order q and let P, Q be generators of respectively $\mathbb{G}_1$ and $\mathbb{G}_2$. We say that $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ are* ***asymmetric bilinear map groups*** *if there exists a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ satisfying the following properties:*

1. *Bilinearity: $\forall\ (S, T) \in \mathbb{G}_1 \times \mathbb{G}_2,\ \forall\ a, b \in \mathbb{Z}$, we have the relation $e(aS, bT) = e(S, T)^{ab}$.*
2. *Non-degeneracy: $\forall\ S \in \mathbb{G}_1,\ e(S, T) = 1\ \forall\ T \in \mathbb{G}_2$ iff $S = 1_{\mathbb{G}_1}$.*
3. *Computability: $\forall\ (S, T) \in \mathbb{G}_1 \times \mathbb{G}_2,\ \ e(S, T)$ is efficiently computable.*
4. *There exists an efficient, publicly computable (but not necessarily invertible) isomorphism $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$ such that $\psi(Q) = P$.*

Such bilinear map groups are known to be instantiable with ordinary elliptic curves such as MNT curves [25] or a kind of curves studied by Barreto and Naehrig [6].

In [10], the next two problems were formalized in the setting of asymmetric bilinear map groups. We note that their hardness does not depend on the availability of an efficiently computable isomorphism $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$.

Definition 2 ([10]). *The* ***Computational co-Diffie-Hellman (co-CDH)*** *problem in asymmetric bilinear map groups $(\mathbb{G}_1, \mathbb{G}_2)$ is to compute $abP_1 \in \mathbb{G}_1$ given $(P_1, P_2, aP_1, bP_2) \in \mathbb{G}_1 \times \mathbb{G}_2 \times \mathbb{G}_1 \times \mathbb{G}_2$. The* ***Decision co-Diffie-Hellman (co-DDH)*** *problem in asymmetric bilinear map groups $(\mathbb{G}_1, \mathbb{G}_2)$ is, given $(P_1, P_2, aP_1, bP_2, cP_1) \in \mathbb{G}_1 \times \mathbb{G}_2 \times \mathbb{G}_1 \times \mathbb{G}_2 \times \mathbb{G}_1$, to decide whether $c \equiv ab \bmod q$.*

The co-DDH problem is easy in asymmetric bilinear map groups. Indeed, given an instance $(P_1, P_2, aP_1, bP_2, cP_1)$, one can decide whether $c = ab$ by checking if $e(aP_1, bP_2) = e(cP_1, P_2)$. According to the terminology of [26], those groups are called *Gap co-Diffie-Hellman groups*. In this paper, we present a scheme taking advantage of the separation between the co-DDH and co-CDH problems.

3 Security notions for signcryption schemes

Definition 3. *A signcryption scheme is made of four algorithms which are the following.*

Common-Keygen: *is a probabilistic algorithm that takes as input a security parameter to output public information I that will be used by all users of the scheme. Such a public information may include the description of plaintext and ciphertext spaces, public and private key spaces, a set of hash functions employed in the scheme, etc. This algorithm may be run either by an authority or by a designated user.*

Keygen: *is a probabilistic key generation algorithm independently run by each user on input of the common public information I to produce a key pair $(\mathsf{sk}, \mathsf{pk})$.*

Sign/Encrypt: *is a probabilistic algorithm that takes as input the common public key I, a plaintext M, the recipient's public key pk_R, and the sender's private key sk_S, and outputs a ciphertext $\sigma = \mathsf{Sign/Encrypt}(M, \mathsf{sk}_S, \mathsf{pk}_R)$.*

Decrypt/Verify: *is a deterministic algorithm that takes as input a ciphertext σ, public information I and the receiver's private key* sk *to return a plaintext M together with a sender's public key* pk_S *and auxiliary non-repudiation information (allowing to convince a third party of the origin of the message) or a distinguished symbol $\perp$ if σ does not properly decrypt into a message accompanied by suitable authenticating information for the sender.*

For correctness, if $\sigma = \mathsf{Sign/Encrypt}(m, \mathsf{sk}_S, \mathsf{pk}_R)$, *the output of the* $\mathsf{Decrypt/Verify}(.)$ *for the pair* (σ, sk_R) *contains a plaintext m, the public key* pk_S *and additional information allowing the receiver to convince a third party that the plaintext actually emanates from the sender.*

Our model considers schemes having the same interface as a sequential composition in the "sign-then-encrypt" order where the sender's public key is encrypted together with the plaintext. The receiver is not assumed to know in advance who the sender is. The latter's identity is retrieved in the decryption/verification process. This difference with models of [1, 3, 15, 23] is motivated by the compatibility with anonymous ciphertexts. We stress here that we do not claim the above model to be more relevant than those of [1, 3, 23, 15]. We just say that it might be more appropriate for specific applications where it is of interest to have anonymous ciphertexts.

We have to mention that Boyen's work [12] was the first one to extend the notion of key privacy [7] to the signcryption setting. Somewhat surprisingly, it was only considered in the identity based setting. A similar notion was never formalized for traditional public keys. The models described in the next section thus aim at filling this gap.

Definition 4. *We say that a signcryption scheme ensures* ***message privacy against chosen-ciphertext attacks*** *(we call this security notion IND-SC-CCA) if no PPT adversary has a non-negligible advantage in the following game:*

1. *The challenger $\mathcal{CH}$ generates a private/public key pair (sk_U, pk_U). sk_U is kept secret while pk_U is given to the adversary $\mathcal{A}$.*
2. *$\mathcal{A}$ performs a first series of queries of the following kinds:*
   - *Signature/encryption queries: $\mathcal{A}$ produces a message $m \in \mathcal{M}$ and an arbitrary public key pk_R (which may differ from pk_U) and requires the result* $\mathsf{Sign/Encrypt}(m, sk_U, pk_R)$ *of the signature/encryption oracle.*
   - *Decryption/verification queries: $\mathcal{A}$ produces a ciphertext σ and requires the result of* $\mathsf{Decryt/Verify}(\sigma, sk_U)$ *which consists of a signed plaintext together with a sender's public key if the obtained signed plaintext is valid for the recovered sender's public key and the $\perp$ symbol otherwise (indicating that the ciphertext was not properly formed).*

   These queries can be asked adaptively: each query may depend on the answers to previous ones. After a number of queries, $\mathcal{A}$ produces two plaintexts $m_0, m_1 \in \mathcal{M}$ and an arbitrary private key sk_S. $\mathcal{CH}$ flips a coin $b \xleftarrow{R} \{0,1\}$ to compute an authenticated encryption $\sigma = \mathsf{Sign/Encrypt}(m_b, sk_S, pk_U)$ *of m_b with the sender's private key sk_S under the attacked public key pk_U. The ciphertext σ is sent to $\mathcal{A}$ as a challenge.*
3. *$\mathcal{A}$ performs new queries as in step 2 but she may not ask the decryption/verification of the challenge σ. At the end of the game, she outputs a bit b' and wins if $b' = b$.*

$\mathcal{A}$'s advantage is defined to be $Adv^{\mathtt{ind\text{-}cca}}(\mathcal{A}) := 2 \times Pr[b' = b] - 1$.

Similarly to several other works [1, 21, 15, 24], our PKC'04 paper considered the non-repudiation property with respect to the entire ciphertext as formalized by the following definition.

Definition 5. *We say that a signcryption scheme is* ***strongly existentially ciphertext-unforgeable*** *against chosen-message attacks (SC-SUF-CMA) if no PPT adversary has a non-negligible advantage in the following game:*

1. *The challenger generates a key pair (sk_U, pk_U) and pk_U is given to the forger $\mathcal{F}$.*
2. *The forger $\mathcal{F}$ performs queries to oracles $\mathsf{Sign/Encrypt}(., sk_U, .)$ and $\mathsf{Decrypt/Verify}(., sk_U)$ exactly as in the previous definition. Again, these queries can also be produced adaptively.*
3. *Eventually, $\mathcal{F}$ produces a ciphertext σ and a key pair (sk_R, pk_R) and wins the game if the result $\mathsf{Decrypt/Verify}(\sigma, sk_R)$ is a tuple (m, s, pk_U) such that (m, s) is a valid signature for the public key pk_U such that σ was not the output of a signature/encryption query $\mathsf{Sign/Encrypt}(m, sk_U, pk_R)$ made during the game.*

As in [1, 21, 15] and many other works, the forger is allowed to have obtained the forged ciphertext as the result of a signature/encryption query for a different receiver's public key than the one corresponding to the claimed forgery. The only constraint is that, for the message m obtained by decryption/verification of the alleged forgery with the chosen private key sk_R, the produced ciphertext σ was not obtained as the result of a $\mathsf{Sign/Encrypt}(m, sk_U, pk_R)$ query.

In [12], Boyen proposed additional security notions for signcryption schemes. One of the most important ones was the notion of ciphertext anonymity that can be viewed as an extension to authenticated encryption schemes of the notion of key privacy already considered by Bellare et al in [7]. Intuitively, in the context of public key encryption, a scheme is said to have the key privacy property if ciphertexts convey no information about the public key that was used to create them. In the signcryption setting, we say that the ciphertext anonymity (or key privacy) property is satisfied if ciphertexts contain no information about who created them nor about whom they are intended to. This notion is a transposition into the non-identity based setting of the one presented by Boyen in [12].

Definition 6. *We say that a signcryption scheme satisfies the **ciphertext anonymity property** (also called key privacy or key indistinguishability: we call this notion SC-INDK-CCA for short) if no PPT distinguisher has a non-negligible advantage in the following game:*

1. *The challenger generates two distinct key pairs $(sk_{R,0}, pk_{R,0})$ and $(sk_{R,1}, pk_{R,1})$. The distinguisher $\mathcal{D}$ is provided with $pk_{R,0}$ and $pk_{R,1}$.*
2. *$\mathcal{D}$ adaptively performs queries $\mathsf{Sign/Encrypt}(m, sk_{R,c}, pk_R)$, for arbitrary receivers' public keys pk_R, and $\mathsf{Decrypt/Verify}(\sigma, sk_{R,c})$ for $c = 0$ or $c = 1$. Once stage 2 is over, $\mathcal{D}$ outputs two private keys $sk_{S,0}$ and $sk_{S,1}$ and a plaintext $m \in \mathcal{M}$. The challenger then flips two coins $b, b' \leftarrow_R \{0, 1\}$ and computes a challenge ciphertext $\sigma = \mathsf{Sign/Encrypt}(m, sk_{S,b}, pk_{R,b'})$ for $\mathcal{D}$.*
3. *$\mathcal{D}$ adaptively performs new queries as in stage 2 with the restriction that, this time, she is disallowed to ask for the decryption/verification of the challenge σ for the private keys $sk_{R,0}$ or $sk_{R,1}$. At the end of the game, $\mathcal{D}$ outputs bits d, d' and wins if $(d, d') = (b, b')$. Her advantage is defined to be $Adv^{\mathsf{indk\text{-}cca}}(D) := Pr[(d, d') = (b, b')] - \frac{1}{4}$.*

Again, this notion captures the security against insider attacks. Indeed, the distinguisher is allowed to choose the pair of private keys among which the one used to create the challenge ciphertext is picked by the challenger. The above definition can be viewed as a transposition to the non-identity based setting of the definition of ciphertext anonymity proposed by Boyen [12] as well as an extension of the definition of key privacy [7] to the authenticated encryption context.

We introduce another notion called 'key invisibility' which is close to the concept (formalized by Galbraith and Mao in [18]) of invisibility for undeniable signatures. Intuitively, this notion expresses the impossibility to decide whether a given ciphertext was actually created using a given particular sender's private key and a given particular receiver's public key.

Definition 7. *We say that a signcryption scheme satisfies the key invisibility (we denote this notion by SC-INVK-CCA for short) if no PPT distinguisher has a non-negligible advantage in the following game:*

1. *The challenger generates a private/public key pair* (sk_U, pk_U). pk_U *is given to the distinguisher* $\mathcal{D}$.
2. $\mathcal{D}$ *adaptively performs queries* Sign/Encrypt(m, sk_U, pk_R), *for arbitrary recipient keys* pk_R, *and* Decrypt/Verify(σ, sk_U).
3. *Once stage 2 is over,* $\mathcal{D}$ *outputs a private key* sk_S *and a plaintext* $m \in \mathcal{M}$. *The challenger then flips a coins* $b \leftarrow_R \{0,1\}$. *If* $b = 0$, *then the challenger returns an actual challenge ciphertext* $\sigma =$ Sign/Encrypt(m, sk_S, pk_U) *to* $\mathcal{D}$. *If* $b = 1$, *then the challenger returns a random* σ *uniformly taken from the ciphertext space* $\mathcal{C}$.
4. $\mathcal{D}$ *adaptively performs new queries as in stage 2 with the restriction that, this time, it cannot require the de-signcryption of the challenge* σ *with the private keys* sk_U.
5. *At the end of the game,* $\mathcal{D}$ *outputs bits* d *and wins if* $d = b$. *Its advantage is defined as*

$$Adv^{\textit{invk-cca}}(D) := 2Pr[d = b] - 1.$$

Again, we allow the distinguisher to choose which private key is used as a part of the challenge to take insider attacks into account.

Galbraith and Mao [18] showed that anonymity and invisibility are essentially equivalent security notions for undeniable signatures. While one can prove in the same way that key privacy and key invisibility are also essentially equivalent for some particular encryption schemes, such an equivalence turns out to be unclear in the signcryption case. In fact, one cannot prove that a distinguisher against the key invisibility implies a distinguisher against the key privacy with the same advantage (because two random coins are used by the challenger in the definition of key privacy and a single one for key anonymity). However, we prove in appendix that, for signcryption schemes satisfying some particular properties (that is, for a given message and a given sender's private key, the output of the signcryption algorithm must be uniformly distributed in the ciphertext space when the receiver's public key is random), we can prove that key invisibility implies key privacy. In the next section we propose a scheme that satisfies both of them (in addition to the more usual notions of message privacy and unforgeability) in the random oracle model.

4 A Key-Private Diffie-Hellman-based Signcryption Scheme

Our solution relies on the digital signature algorithm of Boneh, Lynn and Shacham [10] where signatures have the form $\sigma = xH(m) \in \mathbb{G}_1$ (where the hash function H maps arbitrary messages onto the cyclic group $\mathbb{G}_1$) for public keys $Y = xP_2 \in \mathbb{G}_2$ and can be verified by checking that $e(\sigma, P_2) = e(H(m), Y)$. In order to enhance the concrete security of the reduction in the proof of ciphertext unforgeability, a random quantity U that is used for encryption purposes also acts as a random salt to provide a tighter security reduction to the co-Diffie-Hellman problem in $(\mathbb{G}_1, \mathbb{G}_2)$.

The scheme may be viewed as a composition of a digital signature scheme which is existentially unforgeable against chosen-message attacks (EUF-CMA) [20] with a public key encryption scheme that is only secure against chosen-plaintext attacks. In [1], An, Dodis and Rabin showed that, in a multi-user setting, a composition (in any order) of an EUF-CMA signature with an IND-CCA public key encryption scheme yields a signcryption scheme that is secure in a generalized model of security against chosen-ciphertext attacks provided some precautions are taken (as a mere encrypt-then-sign composition is insecure in the multi-user setting). This section gives an example of secure composition of an EUF-CMA signature with an IND-CPA encryption scheme that yields a CCA-secure signcryption system in the sense of definition 4 instead of a relaxed model of CCA-security. Such a composition is not always secure in general but our example shows that it can be in some particular settings. As we will see, our scheme is secure because redundancies needed to achieve the CCA-security are embedded in the signature.

The version of the scheme that is presented here is not the original one depicted in [22]. The latter version of the scheme was showed to suffer from a slight security flaw that prevents it from being CCA-secure against insider attacks as pointed out in [36]. However, repairing the original scheme was quite straightforward. The version of the system that we describe here is slightly different from the one of [36]. It is also described in terms of asymmetric pairings and may thus be implemented with ordinary curves (we recall that the symbol $\oplus$ denotes the bitwise exclusive OR).

Common-Keygen: given a security parameter k, this algorithm outputs a k-bit prime q, asymmetric bilinear map groups $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ of order q such that the lengths ℓ_1 and ℓ_2 of elements from $\mathbb{G}_1$ and $\mathbb{G}_2$ are polynomial in k, an efficiently computable (but non-necessarily invertible) isomorphism $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$, generators $P_2 \in \mathbb{G}_2$, $P_1 = \psi(P_2)$ and hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$, $H_2 : \{0,1\}^* \rightarrow \{0,1\}^{\ell_1}$ and $H_3 : \{0,1\}^* \rightarrow \{0,1\}^{\lambda}$ where n denotes the size of plaintexts (i.e. the message space is $\mathcal{M} = \{0,1\}^n$). A symmetric encryption scheme (E, D) of keylength λ is also chosen. The common key is then

$$I = \{q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, P_1, P_2, e, H_1, H_2, E, D, n, \ell_1, \ell_2\}.$$

Keygen: user u picks a random $x_u \xleftarrow{R} \mathbb{Z}_q^*$ and sets his public key to $Y_u = x_u P_2 \in \mathbb{G}_2$. His private key is x_u. We will denote the sender and the receiver respectively by $u = S$ and $u = R$ and their key pair by (x_S, Y_S) and (x_R, Y_R).

Sign/Encrypt: given a plaintext $m \in \{0,1\}^n$ intended to R, the sender S uses the following procedure

1. Pick a random $r \xleftarrow{R} \mathbb{Z}_q^*$ and compute $U = rP_1 \in \mathbb{G}_1$.
2. Compute $V = x_S H_1(m||U||Y_R) \in \mathbb{G}_1$ and scramble it into $W = V \oplus H_2(U||r\psi(Y_R))$.
3. Compute $\kappa = H_3(U||V||r\psi(Y_R)) \in \{0,1\}^{\lambda}$ and then $Z = E_\kappa(m||Y_S) \in \{0,1\}^{n+\ell_1+\ell_2}$.

The ciphertext is given by $\sigma = \langle U, W, Z\rangle \in \mathbb{G}_1 \times \{0,1\}^{n+\ell_1+\ell_2}$.

Decrypt/Verify when receiving a ciphertext $\sigma = \langle U, W, Z\rangle$, the receiver R has to perform the steps below:

1. Compute $V = W \oplus H_2(U||x_R U)$ and return $\perp$ if $V \notin \mathbb{G}_1$.
2. Compute $\kappa = H_3(U||V||x_R U) \in \{0,1\}^{\lambda}$.
3. Compute $(m||Y_S) = D_\kappa(Z) \in \{0,1\}^{n+\ell_1+\ell_2}$.
4. Compute $H = H_1(m||U||Y_R) \in \mathbb{G}_1$ and then check if $e(H, Y_S) = e(V, P_2)$. If this condition does not hold, reject the ciphertext.

The consistency of the scheme is easy to verify. We note that, in the signature/encryption algorithm, the recipient's public key must be hashed together with the pair (m, U) in order to achieve the provable strong unforgeability.

When the recipient wishes to convince a third party that a ciphertext actually emanates from the sender, he has to forward m, U, the detached signature V and his public key Y_R to that third party. As public keys lie in group $\mathbb{G}_2$ whose elements have a large representation, it might be more convenient to implement the scheme with symmetric pairings (and thus supersingular cuves) to reduce the communication cost of the non-repudiation procedure in environments where bandwidth is a primary concern.

From an efficiency point of view, only three scalar multiplications in $\mathbb{G}_1$ are required for the signature-encryption operation while 1 multiplication and 2 pairings must be performed in the decryption/verification process. We can verify that the scheme is at least as efficient and more compact than any sequential composition of the BLS signature [10] with any one of the CCA-secure Diffie-Hellman based encryption schemes proposed in [2, 13, 16, 17, 28, 32]. For example, a sequential combination of the BLS signature with the encryption scheme proposed in [2] would involve an additional multiplication at decryption because of the re-encryption phase entailed by the validity checking of the decryption algorithm. If we take $\ell_1 \approx k \geq 160$ and $\ell_2 = 1024$ (by working with an ordinary curve), we see that ciphertexts are about 1344 bits longer than plaintexts. A composition of the BLS signature with the so-called 'length-saving ElGamal encryption scheme' proposed in [2] (which is nothing but the result of applying the Fujisaki-Okamoto transformation of [17] to a variant of ElGamal) would result in ciphertexts that would be at least 1504 bits longer than plaintexts.

We observe that the scheme looks like a sequential composition of the BLS signature with the Hybrid ElGamal encryption scheme proven secure by Cramer and Shoup [14]. Actually, the scheme is more than just a sequential composition. Indeed, the Hybrid ElGamal scheme *must* be implemented with an IND-CCA symmetric encryption (and is trivially CCA-insecure if instantiated with an IND-CPA symmetric scheme). In contrast, our system only needs a symmetric scheme that satisfies the very weak requirement to be semantically secure against passive attacks (that is an attack where the adversary has no encryption or decryption oracle in an indistinguishability scenario). For example, the symmetric encryption could simply be a "one-time pad" with a hash value of the triple $(U, Y_R, r\psi(Y_R))$. The scheme may actually be regarded as a sequential composition of an existentially unforgeable signature with a public key encryption scheme that is only secure against chosen-plaintext attacks. All the redundancies needed to ensure the security in the sense of definition 4 are contained in the signature embedded in the ciphertext. In [1], it was already observed that a sequential composition in the "sign-then-encrypt" order can amplify rather than simply preserve the security properties of the underlying signature and encryption building blocks. Our co-Diffie-Hellman based construction gives another example of this fact.

5 Security Analysis

The original version [22] of this system (where κ was obtained by hashing v alone) was found [35, 36] not to meet its intended security properties. Although a chosen-ciphertext attack was also given [34] against the modification suggested in [36], its present variant is immune to attacks reported in [36, 35, 34] (and countermeasures do not incur any significant additional cost).

In the random oracle model, the scheme is secure under tight reductions assuming the hardness of a natural variant of the Diffie-Hellman problem.

Theorem 1. *Assume that an adversary $\mathcal{A}$ has non-negligible advantage ϵ over the IND-SC-CCA security of the above scheme when running in time t and performing q_{se} signature/encryption queries, q_{dv} decryption/verification queries and q_{H_i} queries to oracles H_i (for $i = 1, 2$). Then, for any $0 \leq \nu \leq \epsilon$, there either exists*

- *an algorithm $\mathcal{B}$ that can solve the co-CDH problem in groups $(\mathbb{G}_1, \mathbb{G}_2)$ with probability $\epsilon' \geq \epsilon - \nu - q_{dv}/2^k$ within time*

$$t' < t + O(q_{dv} + q_{H_2} + q_{H_3})t_p$$

  where t_p denotes the time required for a pairing evaluation.
- *a passive adversary breaking the semantic security of the symmetric scheme (E, D) with advantage ν within time t'.*

Proof. We show how to build an algorithm $\mathcal{B}$ that runs the attacker $\mathcal{A}$ as a subroutine to solve the co-Diffie-Hellman problem in a polynomial time. Let $(aP_1, bP_2) \in \mathbb{G}_1 \times \mathbb{G}_2$ be a random instance of the co-CDH problem. Algorithm $\mathcal{B}$ starts $\mathcal{A}$ with $Y_u = bP_2 \in \mathbb{G}_2$ as a challenge public key. $\mathcal{A}$ then adaptively performs queries that are handled using lists L_1, L_2 and L'_2 to keep track of the answers given to random oracle queries on H_1 and H_2.

- H_1-queries: when a hash query $H_1(m||U||Y_R)$ is made, $\mathcal{B}$ first looks if the value of H_1 was previously defined for the input $m||U||Y_R$. If it was, the previously defined value is returned. Otherwise, $\mathcal{B}$ picks a random $t \xleftarrow{R} \mathbb{Z}_q^*$, returns $tP_1 \in \mathbb{G}_1$ as an answer and inserts the tuple (m, U, Y_R, t) into L_1.
- H_2-queries on inputs $U_i||R_i \in \mathbb{G}_1^2$: $\mathcal{B}$ checks if (P_2, U_i, Y_u, R_i) is a valid co-Diffie-Hellman quadruple (in our notation, we write $R_i = \text{co-DH}_{P_2}(U_i, Y_u)$) by checking whether $e(R_i, P_2) = e(U_i, Y_u)$. If it is then
  - if $U = aP_1$, $\mathcal{B}$ halts and outputs $R_i = abP_1 \in \mathbb{G}_1$ which is the solution that it was looking for.
  - $\mathcal{B}$ checks if L_2' contains an entry of the shape $(U_i, ., h_{2,i})$ for some $h_{2,i} \in \{0,1\}^{\ell_1}$. In this case, $h_{2,i}$ is returned and a record $(U_i, R_i, h_{2,i}, 1)$ is stored in L_2. If no entry $(U_i, ., h_{2,i})$ exists in L_2', $\mathcal{B}$ returns a random string $h_{2,i} \xleftarrow{R} \{0,1\}^{\ell_1}$ and inserts $(U_i, R_i, h_{2,i}, 1)$ in L_2.

  If (P_2, U_i, Y_u, R_i) is not a co-DH tuple, the simulator stores the 5-uple $(U_i, R_i, h_{2,i}, 0)$ in L_2.
- H_3-queries on triples $U_i||V_i||R_i$: $\mathcal{B}$ proceeds as for answering H_2-queries, using lists L_3 and L_3' to maintain the consistency and checking if (P_2, U_i, Y_u, R_i) is a co-DH tuple. Again, the simulator halts and outputs R_i if a co-DH tuple (aP_1, V_i, R_i) is queried.
- For a signature/encryption query on a plaintext m with a recipient's public key Y_R both chosen by $\mathcal{A}$, $\mathcal{B}$ first picks a random $r \xleftarrow{R} \mathbb{Z}_q$, computes $U = rP_1 \in \mathbb{G}_1$ and checks if L_1 contains a tuple (m, U, Y_R, t) indicating that $H_1(m||U||Y_R)$ was previously set to tP_1. If no such tuple is found, $\mathcal{B}$ picks $t \xleftarrow{R} \mathbb{Z}_q$ and stores the entry (m, U, Y_R, t) in L_1. It then computes $V = t\psi(Y_u) = t(bP_1) \in \mathbb{G}_1$. The rest follows as in the normal process: $\mathcal{B}$ computes $r\psi(Y_R)$ (for the Y_R specified by the adversary), simulates H_2 and H_3 to obtain $h_2 = H_2(U||r\psi(Y_R))$, $\kappa = H_3(U||V||r\psi(Y_R))$ and then computes $W = V \oplus h_2$ and $Z = E_\kappa(m||Y_u)$. The ciphertext $\langle U, W, Z\rangle$ is then returned.
- Decryption/verification queries for a ciphertext $C = \langle U, W, Z\rangle$: $\mathcal{B}$ checks if L_2 contains the sole possible 5-uple $(U, Y_u, R, h_2, 1)$ for some elements $R \in \mathbb{G}_1$ and $h_2 \in \{0,1\}^{\ell_1}$ (meaning that $R = \text{co-DH}_{P_2}(U, Y_u)$ and that $H_2(U||R)$ was set to $h_2 \in \{0,1\}^\lambda$):
  - if it does, $\mathcal{B}$ obtains $V = Z \oplus h_2$ and rejects C if $V \notin \mathbb{G}_1$. Otherwise, $\mathcal{B}$ obtains $\kappa = H_3(U||V||R) \in \{0,1\}^\lambda$ (by simulating H_3), sets $(m||Y_S) = D_\kappa(Z)$ and rejects C if $Y_S \notin \mathbb{G}_2$. Otherwise, it computes $H = H_1(m||U||Y_R) \in \mathbb{G}_1$ and checks whether $e(V, P_2) = e(H, Y_S)$. If yes, the information (m, V, Y_S, Y_R) is returned. Otherwise, C is declared invalid.
  - if not, $\mathcal{B}$ draws $h_2 \xleftarrow{R} \{0,1\}^{\ell_1}$ and stores a record $(U, ., h_2)$ in L_2' so as to answer h_2 to a subsequent H_2-query on the input $U||\text{co-DH}_{P_2}(U, Y_u)$. It sets $V = W \oplus h_2 \in \{0,1\}^{\ell_1}$ (and rejects C if $V \notin \mathbb{G}_1$) and inserts a record $(U, V, ., \kappa)$ in L_3' so that a future hash query $H_3(U||V||\text{co-DH}_{P_2}(U, Y_u))$ will get κ as an answer. Finally, $D_\kappa(Z)$ is computed and parsed into $(m||Y_S) \in \{0,1\}^{n+\ell_2}$. The ciphertext C is declared invalid if $Y_S \notin \mathbb{G}_2$ or $e(V, P_2) \neq e(H, Y_S)$ where $H = H_1(m||U||Y_R)$.

At the end of the first stage, $\mathcal{A}$ outputs plaintexts m_0 and m_1 together with an arbitrary sender's private key x_S and requires a challenge ciphertext built under the recipient's public key Y_u. $\mathcal{B}$ then picks a random bit $d \xleftarrow{R} \{0,1\}$ and strings $\kappa^* \xleftarrow{R} \{0,1\}^\lambda$, $W \xleftarrow{R} \{0,1\}^{\ell_1}$. It obtains $H^* = H_1(m_d||aP_1||Y_u) \in \mathbb{G}_1$ by simulating H_1 to compute the challenge ciphertext $\sigma = \langle U, W, Z\rangle = \langle aP_1, W, E_{\kappa^*}(m_d||x_S P_2)\rangle$ which is sent to $\mathcal{A}$. The latter then issues a second series of queries which are handled as in the first stage. If the symmetric scheme (E, D) is semantically secure against passive attacks, it is easy to show that $\mathcal{A}$'s view is independent from the hidden bit $d \in \{0,1\}$ unless she queries a tuple containing abP_1 to random oracles H_2 or H_3. In either case, the solution of the co-Diffie-Hellman problem is detected when answering H_2 and H_3-queries.

Now to assess $\mathcal{B}$'s probability of success, let us denote by AskH the event that $\mathcal{A}$ asks the hash value of abP_1) during the simulation. As done in several papers in the literature (see [32, 9, 12] for instance), as long as the simulation of the attack's environment is perfect, the probability for AskH to happen is the same as in a real attack (i.e. an attack where $\mathcal{A}$ interacts with real oracles). In a real attack we have

$$\begin{aligned}\Pr[b = b'] &\leq \Pr[b = b'|\neg\mathsf{AskH}]\Pr[\neg\mathsf{AskH}] + \Pr[\mathsf{AskH}]\\ &\leq \frac{\nu+1}{2}(1 - \Pr[\mathsf{AskH}]) + \Pr[\mathsf{AskH}]\\ &\leq \frac{1-\nu}{2}\Pr[\mathsf{AskH}] + \frac{\nu+1}{2} \leq \frac{1}{2}\Pr[\mathsf{AskH}] + \frac{\nu+1}{2}\end{aligned}$$

where ν denotes the maximal advantage of any passive adversary against the semantic security of the symmetric scheme (E, D). Since by definition, $\epsilon = 2 \times \Pr[b = b'] - 1$, we may write $\Pr[\mathsf{AskH}] \geq \epsilon - \nu$.

We note that the simulation only fails if the random challenge ciphertext was submitted to the $\mathsf{Decrypt/Verify}$ oracle before the challenge phase. Such a very unlikely event happens with probability smaller than $q_{dv}/2^k$. The bound on $\mathcal{B}$'s computation time derives from the fact that decryption/verification queries and H_2-queries each require 2 pairing evaluations. □

We observe that the reduction is very tight in that, up to negligible terms, algorithm $\mathcal{B}$ has the same probability to solve the co-CDH problem as the adversary's advantage. Moreover, the cost of the reduction is bounded by an expression which is linear in the number of adversarial queries. That is the reason why U is included among the arguments of H_2. The scheme remains secure is V is concealed by a hash value of $r\psi(Y_R)$ alone but the reduction then entails a number of pairing evaluation that is quadratic in the number of adversarial queries (more precisely, up to $2q_{h_2}$ pairings might have to be calculated for each decryption/verification queries).

The reductions are also efficient in the proofs of ciphertext unforgeability and anonymity.

Theorem 2. *Let an adversary $\mathcal{F}$ having non-negligible advantage ϵ over the SC-SUF-CMA security of the scheme when running in time t, making q_{se} signature/encryption queries, q_{dv} decryption/verification queries and q_{H_i} queries on oracles H_i (for $i = 1, 2$). Then there is an algorithm $\mathcal{B}$ solving the co-CDH problem in $(\mathbb{G}_1, \mathbb{G}_2)$ with probability*

$$\epsilon' > \epsilon - (q_{se}(q_{H_1} + q_{se}) + 1)/2^k$$

in time $t' < t + O(q_{H_2} + q_{H_3} + q_{dv})t_p$, t_p being the time required for a pairing evaluation.

Proof. $\mathcal{B}$ receives a random instance (aP_1, bP_2) of the co-Diffie-Hellman problem. It uses $\mathcal{F}$ as a subroutine to solve that instance and plays the role of $\mathcal{F}$'s challenger. The forger $\mathcal{F}$ is initialized with $Y_u = bP_2$ as a challenge public key. She then performs adaptive queries that are handled as explained below (using lists L_1, L_2 and L_2' as in the proof of theorem 1):

- H_2-queries: are dealt with as in the proof of theorem 1.
- H_1-queries on a tuple $m||U||Y_R$: if the latter was previously queried, $\mathcal{B}$ returns the defined value. For a query on a new tuple $m||U||Y_R$, $\mathcal{B}$ picks $t \stackrel{R}{\leftarrow} \mathbb{Z}_q^*$ and defines $H_1(m||U||Y_R) = t(aP_1) \in \mathbb{G}_1$. The list L_1 is updated accordingly.
- Signature/encryption queries for a message m and a receiver's public key Y_R chosen by $\mathcal{F}$: $\mathcal{B}$ picks $r \stackrel{R}{\leftarrow} \mathbb{Z}_q^*$, computes $U = rP_1 \in \mathbb{G}_1$. If H_1 is already defined on $m||U||Y_R$, $\mathcal{B}$ outputs "failure" and halts. Otherwise, $\mathcal{B}$ picks $t \stackrel{R}{\leftarrow} \mathbb{Z}_q^*$, sets $H_1(m||U||Y_R) = tP_1 \in \mathbb{G}_1$ and updates L_1 accordingly. It then computes $V = t\psi(Y_u) \in \mathbb{G}_1$, $h_2 = H_2(U||r\psi(Y_R)) \in \{0,1\}^\lambda$, $W = V \oplus h_2 \in \{0,1\}^{\ell_1}$, $\kappa = H_3(U||V||r\psi(Y_R)) \in \{0,1\}^\lambda$ and $Z = E_\kappa(m||Y_u) \in \{0,1\}^{n+\ell_2}$. The ciphertext $\langle U, W, Z\rangle$ is then returned to $\mathcal{F}$.

- Decryption/verfication queries: are handled exactly as in the proof of theorem 1.

At the end of the game, $\mathcal{F}$ produces a ciphertext $\langle U^*, W^\star, Z^*\rangle$ and a recipient's public/private key pair (x_R^*, Y_R^*). At that moment, $\mathcal{B}$ can perform the decrytion/verification operation using x_R^* and, if the ciphertext is a valid forged ciphertext for the sender's public key Y_u, $\mathcal{B}$ can extract the message-key pair $(m^*||Y_u)$ and the signature V^*. If the hash value $H_1(m^*||U^*||Y_R^*)$ was not asked by $\mathcal{F}$ during the simulation, $\mathcal{B}$ reports "failure" and stops. Otherwise, $H_1(m^*||U^*||Y_R^*)$ must have been set to $t^*(aP_1)$, for some known $t^* \in \mathbb{Z}_q^*$, and V' must be equal to $t^*(abP_1)$ which yields the co-Diffie-Hellman solution $t^{*-1}V'$. It is easy to see that the probability for $\mathcal{B}$ to fail in answering a signature/encryption query is not greater than $q_{se}(q_{h_1} + q_{se})/2^k$ (since at each signature/encryption query, there is at most $q_{H_1} + q_{se}$ elements in L_1). Besides, the probability that $\mathcal{F}$ succeeds in her attempt without asking the query $H_1(m^*, U^*, Y_R^*)$ is at most $1/2^k$. □

Theorem 3. *Assume there exists a PPT distinguisher $\mathcal{D}$ that has non-negligible advantage over the SC-INDK-CCA security of the scheme when running in time t, performing q_{se} signature/encryption queries, q_{dv} decryption/verification queries and q_{H_i} queries to oracle H_i (for $i = 1, 2$). Then, for any $0 \le \nu \le \epsilon$, there either exists*

- *an algorithm $\mathcal{B}$ that solves the co-CDH problem with advantage $\epsilon' > \epsilon - \nu - \frac{q_{dv}}{2^k}$ within time $t' < t + O(q_{dv} + q_{H_2} + q_{H_3})t_p$ where t_p denotes the time required for a pairing calculation.*
- *a passive attacker breaking the semantic security of (E, D) with advantage $2\nu - \frac{1}{2}$ within time t'.*

Proof. Let (aP_1, bP_2) be an instance of the co-CDH problem. $\mathcal{B}$ uses $\mathcal{A}$ to solve that instance and plays the role of $\mathcal{D}$'s challenger in the game of definition 6. To do this, $\mathcal{B}$ picks random elements $x, y \in \mathbb{Z}_q^*$ and initializes $\mathcal{D}$ with $pk_{u,0} = Y_{u,0} = x(bP_2)$ and $pk_{u,1} = Y_{u,1} = y(bP_2)$. The distinguisher $\mathcal{D}$ then issues queries as explained in definition 6. To deal with these queries, $\mathcal{B}$ maintains lists L_1, L_2 and L_2' as in previous theorems. When oracle H_2 is queried, $\mathcal{B}$ performs the same manipulations as in previous proofs to maintain consistency with decryption/verification queries. All queries are treated exactly in the same way as in the proof of theorem 1 except that two distinct private keys $Y_{u,0}$ and $Y_{u,1}$ may be involved in calls to oracles $\mathsf{Sign/Encrypt}$ and $\mathsf{Decrypt/Verify}$. It can be checked that it does not change anything to the strategy of the simulator that never fails.

Once the first stage is over, $\mathcal{D}$ outputs two private key $sk_{S,0} = x_{S,0} \in \mathbb{Z}_q^*$, $sk_{S,1} = x_{S,1} \in \mathbb{Z}_q^*$ and a plaintext $m \in \{0,1\}^n$. $\mathcal{B}$ then chooses two independent bits $(d, d') \in \{0,1\} \times \{0,1\}$ and random strings $\kappa \xleftarrow{R} \{0,1\}^\lambda$, $W \xleftarrow{R} \{0,1\}^{\ell_1}$, obtains $H = H_1(m||aP||Y_{u,d'}) \in \mathbb{G}_1$ and computes $Z = E_\kappa(m||x_{S,d}P_2) \in \{0,1\}^{n+\ell_2}$ before sending the challenge $\sigma = \langle aP, W, Z\rangle$ to $\mathcal{D}$. Clearly, if (E, D) is a semantically secure symmetric scheme, as long as no H_2 or H_3-query is made on a tuple containing $x(abP_1)$ or $y(abP_1)$, $\mathcal{F}$'s view is independent from the hidden bits (d, d'). Moreover, such H_2 or H_3-queries would provide $\mathcal{B}$ with the co-CDH solution abP_1 that would be detected when handling hash queries.

$\mathcal{D}$'s second series of queries is handled as in the first stage. Eventually, $\mathcal{D}$ outputs a guess (d, d') which is ignored. We note that in a real attack, we have

$$\begin{aligned}
\Pr[(d,d') = (b,b')] &\le \Pr[(d,d') = (b,b')|\neg\mathsf{AskH}]\Pr[\neg\mathsf{AskH}] + \Pr[\mathsf{AskH}] \\
&= (\nu + \frac{1}{4})(1 - \Pr[\mathsf{AskH}]) + \Pr[\mathsf{AskH}] \\
&\le \nu + \frac{1}{4} + \frac{3}{4}\Pr[\mathsf{AskH}]
\end{aligned}$$

where ν is the maximal advantage of a passive adversary against the symmetric scheme in distinguishing between the encryptions of four messages of her choosing (this advantage being defined as the probability of a correct guess minus 1/4). Indeed, the second equality derives from the fact that $\Pr[(d,d')=(b,b')|\neg\mathsf{AskH}]$ is nothing but the probability of a passive adversary to distinguish between the encryption of four plaintexts for an unknown random symmetric key. As $\Pr[(d,d')=(b,b')]=\epsilon+1/4$ by definition, we thus find $\Pr[\mathsf{AskH}]\geq\frac{4}{3}(\epsilon-\nu)>(\epsilon-\nu)$. As in theorem 1, one can show that, as long as the simulation is perfect, the probability for AskH to happen is the same in the simulation as in a real attack and as in theorem 1, the simulator only fails with negligible probability. Its strategy to solve the co-CDH problem is to check two candidates co-CDH tuples $(aP_1, x(bP_2), .)$ or $(aP_1, y(bP_2), .)$ for each H_2 or H_3-query involving aP_1 and the bound on its running time follows.

The announced bound for the advantage in breaking the semantic security of the symmetric scheme in the event $\neg\mathsf{AskH}$ stems from the fact that an attacker distinguishing between the encryption of 4 messages with probability $\nu+1/4$ implies a distinguisher winning a traditional indistinguishability game with the same probability and thus an advantage $2(\nu+1/4)-1=2\nu-1/2$. □

The proof of invisibility in the sense of definition 7 is very similar to proof of ciphertext anonymity and is omitted here

6 Conclusions

This paper proposed a signcryption construction that enjoys a near-optimal concrete security under a classical number theoretic assumption which is a very rare feature among schemes based on discrete-logarithm-related assumptions. It simultaneously and provably ensures confidentiality, authentication as well as anonymous communications in the random oracle model.

References

1. J.-H. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption. In *Advances in Cryptology - Eurocrypt'02*, LNCS 2332, pp. 83–107. Springer, 2002.
2. J. Baek, B. Lee, and K. Kim. Secure Length-Saving ElGamal Encryption under the Computational Diffie-Hellman Assumption. In *Proceedings of ACISP'00*, LNCS 1841, pp. 49–58. Springer, 2000.
3. J. Baek, R. Steinfeld, and Y. Zheng. Formal Proofs for the Security of Signcryption. In *Proceedings of PKC'02*,LNCS 2274, pp. 80–98. Springer, 2002.
4. J. Baek and Y. Zheng. Simple and Efficient Threshold Cryptosystem from the Gap Diffie-Hellman Group. Available at http://citeseer.nj.nec.com/567030.html.
5. F. Bao and R.-H. Deng. A signcryption scheme with signature directly verifiable by public key. In *Proceedings of PKC'98*, LNCS 1998, pp. 55–59, 1998.
6. P. S. L. M. Barreto and M. Naehrig. Pairing-friendly elliptic curves of prime order. In *SAC'05*. To Appear.
7. M. Bellare, A. Boldyreva, A. Desai, and D. Pointcheval. Key-Privacy in Public-Key Encryption. In *Advances in Cryptology - Asiacrypt'01*, LNCS 2248, pp. 566–582. Springer, 2001.
8. M. Bellare, P. Rogaway, *Random oracles are practical: A paradigm for designing efficient protocols*, Proc. of the 1^{st} ACM Conference on Computer and Communications Security, pp. 62-73, 1993.
9. D. Boneh and M. Franklin. Identity Based Encryption From the Weil Pairing. In *Advances in Cryptology - Proceedings of Crypto'01*, LNCS 2139, pp. 213–229. Springer, 2001.
10. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. In *Advances in Cryptology - Proceedings of Asiacrypt'01*, LNCS 2248, pp. 514–532. Springer, 2001.
11. X. Boyen. Multipurpose identity-based signcryption : A swiss army knife for identity-based cryptography, 2003. extended version of [12], eprint available at http://eprint.iacr.org/2003/163/.
12. X. Boyen. Multipurpose identity-based signcryption: A swiss army knife for identity-based cryptography. In *Advances in Cryptology (CRYPTO '03)*, LNCS 2729, pp. 382–398. Springer, 2003.
13. R. Cramer and V. Shoup. A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. In *Advances in Cryptology - Crypto'98*, LNCS 1462, pp. 13–25. Springer, 1998.
14. R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal of Computing*, 33:167–226, 2003.
15. Y. Dodis, M.-J. Freedman, and S. Walfish. Parallel Signcryption with OAEP, PSS-R and other Feistel Paddings. 2003. Available at http://eprint.iacr.org/2003/043/.

16. E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Advances in Cryptology - Crypto'99*, LNCS 1666, pp. 537–554. Springer, 1999.
17. E. Fujisaki and T. Okamoto. How to Enhance the Security of Public-Key Encryption at Minimum Cost. In *Proceedings of PKC'99*, LNCS 1560, pp. 53–68. Springer, 1999.
18. S. Galbraith and W. Mao. Invisibility and anonymity of undeniable and confirmer signatures. In *Topics in Cryptology - proceedings of CT-RSA 2003*, LNCS 2612, pp. 80-97. Springer, 2003.
19. C. Gamage, J. Leiwo, and Y. Zheng. Encrypted message authentication by firewalls. In *Proceedings of PKC'98*, LNCS 1560, pp. 69–81. Springer, 1998.
20. S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen message attacks. *SIAM Journal of Computing*, 17(2):281–308, 1988.
21. I.-R. Jeong, H.-Y. Jeong, H.-S. Rhee, D.-H. Lee, and I.-L. Jong. Provably secure encrypt-then-sign composition in hybrid signcryption. In *Proceedings of ICISC'02*, LNCS 2587, pp. 16-34. Springer, 2002.
22. B. Libert and J. J. Quisquater. Efficient signcryption with key privacy from gap Diffie-Hellman groups. In *PKC'04*, volume 2947 of *LNCS*, pages 187–200. Springer, 2004.
23. J. Malone-Lee. Signcryption with non-repudiation, 2002. Technical report available at http://www.cs.bris.ac.uk/Tools/Reports/Ps/2002-malonelee.pdf.
24. J. Malone-Lee and W. Mao. Two birds one stone: Signcryption using RSA. In *Topics in Cryptology - proceedings of CT-RSA 2003*, LNCS 2612, pp. 211-225. Springer, 2003.
25. A. Miyaji, M. Nakabayashi, S. Takano. New Explicit Conditions of Elliptic Curve Traces for FR-Reduction. *IEICE Transactions on Fundamentals*, E84-A(5):1234–1243, 2001.
26. T. Okamoto and D. Pointcheval. The Gap-Problems: A New Class of Problems for the Security of Cryptographic Schemes. In *Proceedings of PKC'01*, LNCS 1992. Springer, 2001.
27. J. Pieprzyk and D. Pointcheval. Parallel Authentication and Public-Key Encryption. In *Proceedings of ACISP'03*, LNCS 2727, pp. 383–401. Springer, 2003.
28. D. Pointcheval. Chosen-Ciphertext Security for any One-Way Cryptosystem . In *Proceedings of PKC'00*, LNCS 1751, pp. 129–146. Springer, 2000.
29. D. Pointcheval and J. Stern. Security proofs for signature schemes. In *Advances in Cryptology - Eurocrypt'96*, LNCS 1992, pp. 387–398. Springer, 1996.
30. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. In *Journal of Cryptology*, volume 13-Number 3, pp. 361–396. Springer, 2000.
31. J.-B. Shin, K. Lee, and K. Shim. New DSA-verifiable signcryption schemes. In *Proceedings of ICISC'02*, LNCS 2587, pp. 35–47. Springer, 2002.
32. V. Shoup and R. Gennaro. Securing Threshold Cryptosystems against Chosen Ciphertext Attack. In *Advances in Cryptology - Eurocrypt'98*, LNCS 1403, pp. 1–16. Springer, 1998.
33. R. Steinfeld and Y. Zheng. A signcryption scheme based on integer factorization. In *Proceedings of ISW'00*, LNCS 1975, pp. 308–322, Springer, 2000.
34. C. H. Tan. Analysis of improved signcryption scheme with key privacy. *Infor- mation Processing Letters*, 99:135138, 2006.
35. C. H. Tan. On the security of signcryption scheme with key privacy. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E88-A(4):10931095, 2005.
36. G. Yang, D. S. Wong, and X. Deng. Analysis and improvement of a signcryption scheme with key privacy. In *ISC05*, volume 3650 of LNCS, pages 218232. Springer, 2005.
37. B.-H. Yum and P.-J. Lee. New signcryption schemes based on KCDSA. In *Proceedings of ICISC'01*, LNCS 2288, pp. 305–317, Springer, 2001.
38. Y. Zheng. Digital signcryption or how to achieve cost (signature & encryption) << cost(signature) + cost(encryption). In *Advances in Cryptology - Crypto'97*, LNCS 1294, pp. 165–179, Springer, 1997.

Appendix: proof of the relation between key invisibility and key privacy

We consider a signcryption scheme (K, SC, DSC) made of a key generation algorithm K, a signcryption algorithm SC and a de-signcryption algorithm DSC. We assume the scheme has the following property: for any given m in the space of plaintexts $\mathcal{M}$ and any sk in the space of private keys $\mathcal{SK}$, the distribution $\{\sigma \in \mathcal{C} | \sigma = SC(m, sk, pk) \wedge pk \leftarrow_R \mathcal{PK}\}$ is indistinguishable from a uniform distribution on the space of ciphertexts $\mathcal{C}$, where $\mathcal{PK}$ denotes the space of public keys.

Now we consider, a disinguisher D_A against the key privacy of such a scheme. We show that it implies the existence of a distinguisher D_I against its key invisibility. The proof is inspired from the one of Galbraith and Mao [18] for undeniable signatures. The distinguisher, depicted below, takes as input a public key pk_0 and has access to a challenger $Chall$ such as the one of definition 5.

$D_I^{Chall}(pk_0)$

1. $(pk_1, sk_1) \leftarrow K(k)$
2. $b' \leftarrow_R \{0,1\}$
   If $b' = 0$ send (pk_0, pk_1) to D_A
   else send it (pk_1, pk_0).
3. For any query related to the private key associated to pk_0, forward it to $Chall$ and transmit the answer to D_A. Answer any query related to pk_1 with the private key sk_1.
4. For the tuple (m, sk'_0, sk'_1) outputted by D_A, flip a coin $b'' \leftarrow_R \{0,1\}$; send $(m, sk'_{b''})$ to $Chall$.
5. Receive a challenge σ from $Chall$ and forward it to D_A.
6. Handle all queries made by D_A as in step 3.
7. D_A outputs (d', d''). Return 0 if $(d', d'') = (b', b'')$ and 1 otherwise.

Let b be the random bit chosen by D_I's challenger in the game of definition 5 and let ϵ be D_A's advantage in the game of definition 4. From D_I's behaviour, it is clear that, if $b = 0$ (that is if $Chall$ decides to return an actual ciphertext for the private key $sk'_{b''}$ and the public key pk_0), we have $\Pr[(d', d'') = (b', b'')] = \epsilon + 1/4$. Now, if $b = 1$ (that is if $Chall$ decides to return a randomly chosen σ from the ciphertext space), it is clear that, with overwhelming probability, σ is not a valid ciphertext that could be the output of $SC(m, sk'_u, pk_v)$ for $u, v = 0, 1$. Nevertheless, because of the aforementioned assumed property of the scheme, D_A is unable to realize it and it comes that $\Pr[(d', d'') = (b', b'')|b = 1] \leq 1/4$ and we can then write

$$
\begin{aligned}
\Pr[\mathtt{Output}(D_I) = b] &= \Pr[(d', d'') = (b', b'')|b = 0]\Pr[b = 0] + \Pr[(d', d'') \neq (b', b'')|b = 1]\Pr[b = 1] \\
&\geq (\epsilon + \frac{1}{4})\frac{1}{2} + \frac{3}{4}\frac{1}{2} = \frac{\epsilon}{2} + \frac{1}{2}
\end{aligned}
$$

and hence, $\mathtt{Adv}(D_I) = 2\Pr[\mathtt{Output}(D_I) = b] - 1 \geq \epsilon$.

We find that, for a signcryption scheme satisfying the required property, a distinguisher against the key invisibility has an advantage at least as large as the one of a distinguisher against the key privacy.