Hardness of k-LWE and Applications in Traitor Tracing

San Ling[1], Duong Hieu Phan[2], Damien Stehlé[3], and Ron Steinfeld[4]

[1] Division of Mathematical Sciences,
School of Physical and Mathematical Sciences,
Nanyang Technological University, Singapore
[2] Laboratoire LAGA (CNRS, U. Paris 8, U. Paris 13), U. Paris 8
[3] Laboratoire LIP (U. Lyon, CNRS, ENSL, INRIA, UCBL), ENS de Lyon, France
[4] Faculty of Information Technology,
Monash University, Clayton, Australia

Abstract. We introduce the k-LWE *problem*, a Learning With Errors variant of the k-SIS problem. The Boneh-Freeman reduction from SIS to k-SIS suffers from an exponential loss in k. We improve and extend it to an LWE to k-LWE reduction with a polynomial loss in k, by relying on a new technique involving trapdoors for random integer kernel lattices. Based on this hardness result, we present the first algebraic construction of a traitor tracing scheme whose security relies on the worst-case hardness of standard lattice problems. The proposed LWE traitor tracing is almost as efficient as the LWE encryption. Further, it achieves public traceability, i.e., allows the authority to delegate the tracing capability to "untrusted" parties. To this aim, we introduce the notion of *projective sampling family* in which each sampling function is keyed and, with a projection of the key on a well chosen space, one can simulate the sampling function in a computationally indistinguishable way. The construction of a projective sampling family from k-LWE allows us to achieve public traceability, by publishing the projected keys of the users. We believe that the new lattice tools and the projective sampling family are quite general that they may have applications in other areas.
Keywords. Lattice-based cryptography, Traitor tracing, LWE.

1 Introduction

Since the pioneering work of Ajtai [6], there have been a number of proposals of cryptographic schemes with security provably relying on the worst-case hardness of standard lattice problems, such as the decision Gap Shortest Vector Problem with polynomial gap (see the surveys [44, 58]). These schemes enjoy unmatched security guarantees: Security relies on *worst-case* hardness assumptions for problems expected to be *exponentially hard* to solve (with respect to the lattice dimension n), even with quantum computers. At the same time, they often enjoy great asymptotic efficiency, as the basic operations are matrix-vector multiplications in dimension $\widetilde{O}(n)$ over a ring of cardinality $\leq \mathcal{P}oly(n)$. A breakthrough result in that field was the introduction of the Learning With Errors problem (LWE) by Regev [56, 57], who showed it to be at least as hard as worst-case lattice problems and exploited it to devise an elementary encryption scheme. Gentry et al. showed in [27] that Regev's scheme may be adapted so that a master can generate a large number of secret keys for the same public key. As a result, the latter encryption scheme, called dual-Regev, can be naturally extended into a multi-receiver encryption scheme. In the present work, we build traitor tracing schemes from this dual-Regev LWE-based encryption scheme.

Traitor tracing. A traitor tracing scheme is a multi-receiver encryption scheme where malicious receiver coalitions aiming at building pirate decryption devices are deterred by the existence of a tracing algorithm: Using the pirate decryption device, the tracing algorithm can recover at least one member of the malicious coalition. Such schemes are particularly well suited for fighting copyright infringement in the context of commercial content distribution (e.g., Pay-TV, subscription news websites, etc). Since their introduction by Chor et al. [21], much work has been devoted to devising efficient and secure traitor tracing schemes. The most desirable schemes are fully collusion resistant: they can deal with arbitrarily large malicious coalitions. But, unsurprisingly, the most efficient schemes are in the bounded collusion model where the number of malicious users is limited. The first non-trivial fully collusion resistant scheme was proposed by Boneh et al. [14]. However, its ciphertext size is still large ($\Omega(\sqrt{N})$, where N is the total number of users) and it relies on pairing groups of composite order. Very recently, Boneh and Zhandry [16] proposed a fully collusion resistant scheme with poly-log size parameters. It relies on indistinguishability obfuscation [26], whose security foundation remains to be studied, and whose practicality remains to be exhibited. In this paper, we focus on the bounded collusion model. The Boneh-Franklin scheme [11] is one of the earliest algebraic constructions but it can still be considered as the reference algebraic transformation from the standard ElGamal public key encryption into traitor tracing. This transformation induces a linear loss in efficiency, with respect to the maximum number of traitors. The known transformations from encryption to traitor tracing in the bounded collusion model present at least a linear loss in efficiency, either in the ciphertext size or in the private key size [11, 46, 33, 60, 10, 13]. We refer to [29] for a detailed introduction to this rich topic. Also, in Section 2.1, we give a short overview of traitor tracing schemes with their properties, in particular the public traceability.

Our Contributions. We describe the first algebraic construction of a public-key lattice-based traitor tracing scheme. It is semantically secure and enjoys public traceability. The security relies on the hardness of LWE, which is known to be at least as hard as standard worst-case lattice problems [57, 49, 18].

The scheme is the extension, described above, of the dual-Regev LWE-based encryption scheme from [27] to a multi-receiver encryption scheme, where each user has a different secret key. In the case of traitor tracing, several keys may be leaked to a traitor coalition. To show that we can trace the traitors, we extend the LWE problem and introduce the k-LWE problem, in which k hint vectors (the leaked keys) are given out.

Intuitively, k-LWE asks to distinguish between a random vector $\boldsymbol{t}$ close to a given lattice Λ and a random vector $\boldsymbol{t}$ close to the orthogonal subspace of the span of k given short vectors belonging to the dual Λ^* of that lattice. Even if we are given $(\boldsymbol{b}_i^*)_{i\leq k}$ small in Λ^*, computing the inner products $\langle \boldsymbol{b}_i^*, \boldsymbol{t}\rangle$ will not help in solving this problem, since they are small and distributed identically in both cases. The k-LWE problem can be interpreted as a dual of the k-SIS problem introduced by Boneh and Freeman [12], which intuitively requests to find a short vector in Λ^* that is linearly independent with the k given short vectors of Λ^*. Their reduction from SIS to k-SIS can be adapted to the LWE setup, but the hardness loss incurred by the reduction is gigantic. We propose a significantly sharper reduction from LWE_α to k-LWE_α. This improved reduction requires a new lattice technique: the equivalent for kernel lattices of Ajtai's simultaneous sampling of a random q-ary lattice with a short basis [7] (see also Lemma 10). We adapt the Micciancio-Peikert framework from [42] to sampling a Gaussian $X \in \mathbb{Z}^{m\times n}$ along with a short basis for the lattice $\ker(X) = \{\boldsymbol{b} \in \mathbb{Z}^m : \boldsymbol{b}^t X = \boldsymbol{0}\}$.

Our construction of a traitor tracing scheme from k-LWE can be seen as an additive and noisy variant of the (black-box) Boneh-Franklin traitor tracing scheme [11]. While the Boneh-Franklin scheme is transformed from the ElGamal encryption with a linear loss (in the maximum number of traitors) in efficiency, our scheme is almost as efficient as standard LWE-based encryption, as long as the maximum number of traitors is below m/c, where m is the number of LWE samples and c is a constant. The full functionality of black-box tracing in both the Boneh-Franklin scheme and ours are of high complexity as they both rely on the black-box confirmation: given a superset of the traitors, it is guaranteed to find at least one traitor and no innocent suspect is incriminated. Boneh and Franklin left the improvement of the black-box tracing as an interesting open problem. We show that in the lattice setting, the black-box tracing can be accelerated by running the tracing procedure in parallel on untrusted machines. This is a direct consequence of the property of public traceability, i.e., the possibility of running tracing procedure on public information, that our scheme enjoys. We note that almost all traitor tracing systems require that the tracing key must be kept secret. Some schemes [20, 55, 15, 17] achieve public traceability and some others achieve a stronger notion than public traceability, namely non-repudation, but the setup in these schemes require some interactive protocol between the center and each user such as a secure 2-party computation protocol in [53], a commitment protocol in [54], an oblivious polynomial evaluation in [64, 35, 32].

To obtain public traceability and inspired from the notion of projective hash family [23], we introduce a new notion of *projective sampling family* in which each sampling function is keyed and, with a projection of the key on a well chosen space, one can simulate the sampling function in a computationally indistinguishable way. The construction of a set of projective sampling families from k-LWE allows us to publicly sample the tracing signals.

Related works. Our technique can be adapted to the SIS to k-SIS reduction. We thus solve the open question left by Boneh and Freeman of improving their reduction [12]: from an exponential loss in k to a polynomial loss in k. Consequently, their linearly homomorphic and ordinary signature schemes enjoy much better efficiency/security trade-offs.

The Extended-LWE problem, defined in [48] and studied in [8, 18], has a similar flavour to k-LWE. In k-LWE, the adversary is given k short vectors in the dual Λ^* of the lattice Λ under scope and has to tell whether a vector $\boldsymbol{t}$ is close to Λ or close to the orthogonal of the k vectors. In Extended-LWE, the adversary is given a short vector $\boldsymbol{z}$ along with the value $\langle \boldsymbol{z}, \boldsymbol{e} \rangle$ where $\boldsymbol{e}$ denotes the offset between $\boldsymbol{t}$ and Λ. In [39], Libert and Stehlé used our sampling of a random Gaussian integer matrix X together with a short basis of $\ker(X)$, to show that Extended-LWE remains hard if one is given k well-distributed hint vectors $\boldsymbol{z}$ instead of just one. They used it to prove the security of a fully secure functional encryption scheme for inner products.

Open problems.Our proposed lattice-based traitor tracing scheme resists Chosen Plaintext Attacks. There exist traitor tracing schemes that resist Chosen Ciphertext Attacks, such as [11, Sec. 8], but they rely on more traditional hardness assumptions (such as DDH). It seems quite challenging to devise such an IND-CCA-secure scheme under lattice hardness assumptions. Intuitively, in a traitor tracing scheme the users own parts of a master secret (e.g., each user owns a short vector in a shared lattice, or a discrete log representation with respect to a shared set of group elements), and we attempt to prevent traitors from gaining knowledge of more than their share of the secret information. This requirement seems to be in opposition with the underlying design of lattice-based IND-CCA-secure encryption schemes [49, 52, 19, 3, 4], as the receiver uses the full secret information (a short basis of lattice) to verify the well-formedness of the ciphertext it decrypts. It is an interesting open problem to design an IND-CCA-secure lattice-based encryption scheme where different independent secret keys could be used for a common public key.

Our scheme naturally raises the question whether the additional properties and features that are enjoyed by existing traitor tracing schemes can also be achieved using lattice hardness assumptions. Our scheme looks as a good starting point for building an ID-based traitor tracing scheme [1], as it seems compatible with the construction from [3]. Another popular functionality is the possibility of revoking malicious users [47] or broadcasting to any subgroup of the users [25]. Our scheme can be directly adapted to broadcast to a small group of users (the tracing signals Tr_k can be used for sending messages to users that has one of the sub-keys $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$). This possibility of broadcasting to a small set of the sub-keys can be then combined with the complete subtree framework in [45] to deal with revocation (in which each node of the tree is associated with a sub-key $\boldsymbol{x}_i$ and the key of each user is a set of sub-keys on the path from the root to the user). However, this way leads to a combinatorial and inefficient scheme and can only deal with a small set of the revokers (for ensuring the security). An algebraic construction of a lattice-based trace-and-revoke scheme could be an interesting problem.

Remark. This article is the full version of [40], published in the proceedings of CRYPTO 2014. Compared to the proceedings version, the reduction from LWE to k-LWE has been significantly simplified. This simplification also led to an improvement in the number of hints it handles, from $k = o(n)$ to k linear in m.

2 Preliminaries

If x is a real number, then $\lfloor x \rceil$ is the closest integer to x (with any deterministic rule in case x is half an odd integer). All vectors will be denoted in bold. By default, our vectors are column vectors. We let $\langle \cdot, \cdot \rangle$ denote the canonical inner product. For q prime, we let $\mathbb{Z}_q$ denote the field of integers modulo q. For two matrices A, B of compatible dimensions, we let $(A|B)$ and $(A\|B)$ respectively denote the horizontal and vertical concatenations of A and B. For $A \in \mathbb{Z}_q^{m \times n}$, we define $\mathrm{Im}(A) = \{A\boldsymbol{s} : \boldsymbol{s} \in \mathbb{Z}_q^n\} \subseteq \mathbb{Z}_q^m$. For $X \subseteq \mathbb{Z}_q^m$, we let $\mathrm{Span}(X)$ denote the set of all linear combinations of the elements of X. We let $X^\perp$ denote the linear subspace $\{\boldsymbol{b} \in \mathbb{Z}_q^m : \forall \boldsymbol{c} \in X, \langle \boldsymbol{b}, \boldsymbol{c} \rangle = 0\}$. For a matrix $S \in \mathbb{R}^{m \times n}$, we let $\|S\|$ denote the norm of its longest column. If S is full column-rank, we let $\sigma_1(S) \geq \ldots \geq \sigma_n(S)$ denote its singular values. We let $\mathbb{T}$ denote the additive group $\mathbb{R}/\mathbb{Z}$.

If D_1 and D_2 are distributions over a countable set X, their statistical distance $\frac{1}{2}\sum_{x \in X} |D_1(x) - D_2(x)|$ will be denoted by $\Delta(D_1, D_2)$. The statistical distance is defined similarly if X is measurable. If X is of finite weight, we let $U(X)$ denote the uniform distribution over X. For any invertible $S \in \mathbb{R}^{m \times m}$ and $\boldsymbol{c} \in \mathbb{R}^m$, we define the function $\rho_{S,\boldsymbol{c}}(\boldsymbol{b}) = \exp(-\pi \|S^{-1}(\boldsymbol{b} - \boldsymbol{c})\|^2)$. For $S = \sigma I_m$, we write $\rho_{\sigma,\boldsymbol{c}}$, and we omit the subscripts S and $\boldsymbol{c}$ when $S = I_m$ and $\boldsymbol{c} = \boldsymbol{0}$. We let ν_α denote the one-dimensional Gaussian distribution with standard deviation α.

2.1 Traitor tracing

Combinatorial schemes versus algebraic schemes. There are two main approaches for devising a traitor tracing encryption scheme. Many constructions are combinatorial in nature (see [21, 61, 22, 59, 55, 10, 13], among others): They typically combine an arbitrary encryption scheme with a collusion-resistant fingerprinting code. The most interesting property in combinatorial schemes is the capacity of dealing with black-box tracing. However, the efficiency of these traitor tracing schemes is curbed by the large parameters induced by even the best construction of such codes [63]: To resist coalitions of up to t malicious users among N users, the code length is $\ell = \Theta(t^2 \log N)$. Lower bounds with the same dependence with respect to t have been given in [51, 63], leaving little hope of significant improvements.

An alternative approach was initiated by Kurosawa and Desmedt in [36] (whose construction was shown insecure in [62]), and by Boneh and Franklin [11]: The tracing functionality directly stems from the algebraic properties of the encryption scheme. As opposed to the combinatorial approach, this algebraic approach is not generic and requires designing ad hoc encryption schemes. In this paper, we concentrate on the algebraic approach. Prior to this work, all known algebraic traitor tracing schemes relied on variants of the Discrete Logarithm Problem: For instance, the earlier constructions (including [36, 11, 33, 37]) rely on the assumed hardness of the Decision Diffie Hellman problem (DDH), whereas others (including [20, 14, 15, 1, 24]) rely on variants of DDH on groups admitting pairings. The former provide strong security when instantiating with groups for which DDH is expected to be very hard (such as generic elliptic curves over prime fields), whereas the latter achieve improved functionalities while lowering the performance (as a function of the security level).

Public traceability. An important problem on traitor tracing is to handle the case where the tracer is not trusted. In this scenario, the tracing procedure must be run in a way that enables verification of the traitor implication, by a system outsider. The strongest notion for this is non-repudiation: the tracing procedure must produce an undeniable proof of the traitor implication. However, a necessary condition for achieving non-repudiation is that the setup involves some interactive protocol between the center and each user. Indeed, if the center generates all the parameters for the users, then any pirate decoder produced by a collusion of traitors can also be produced by the center and there is no way for the center to trustworthily prove the culpability of the traitors. All the existing schemes enjoying non-repudiation involve complex interactive proofs: a secure 2-party computation protocol in [53], a commitment protocol in [54], an oblivious polynomial evaluation in [64, 35, 32].

When considering the standard setting of non-interactive setup, we cannot get the full strength of non-repudiation, but we can still achieve a weaker but very useful property: public traceability. This notion allows anyone to perform the tracing from the public parameters only and hence the traitor implication can be publicly verified. Moreover, public traceability implies the capacity of delegating the tracing procedure: the tracer can run the tracing procedure in parallel on untrusted machines without leaking any secret information. This can prove crucial for the schemes with high tracing complexity. In fact, there are very few (non-interactive) schemes that achieve this property [55, 15] (some schemes, such as [20, 10, 13], partially achieve: some parts of the tracing procedure can be run publicly). The scheme [55] is generic, based on IPP-codes, and is thus quite impractical. The Boneh-Waters scheme [15] achieves resistance against unbounded coalitions, but has a large ciphertext size of $\Theta(\sqrt{N})$ group elements. All known efficient algebraic schemes are in the bounded collusion model and so far, none of them enjoys public traceability. In this paper, we achieve public traceability without downgrading the efficiency of the proposed sheme.

Public key traitor tracing encryption

A public-key traitor tracing scheme consists of four probabilistic algorithms `Setup`, `Enc`, `Dec` and `Trace`.

- Algorithm `Setup` is run by a trusted authority. It takes as inputs a security parameter λ, a list of users $(\mathcal{U}_i)_{i\le N}$ and a bound t on the size of traitor coalitions. It computes a public key pk, descriptions of the plaintext and ciphertext domains $\mathcal{P}$ and $\mathcal{C}$, secret keys $(sk_i)_{i\le N}$, and a tracing key tk (which may contain the sk_i's and additional data). It publishes $pk, \mathcal{P}$ and $\mathcal{C}$, and sends sk_i to user $\mathcal{U}_i$ for all $i \le N$.
- Algorithm `Enc` can be run by any party. It takes as inputs a public key pk and a plaintext message $M \in \mathcal{P}$. It computes a ciphertext $C \in \mathcal{C}$.
- Algorithm `Dec` can be run by any user. It takes as inputs a secret key sk_i and a ciphertext message $C \in \mathcal{C}$. It computes a plaintext $P \in \mathcal{P}$.
- Algorithm `Trace` is explained below. If the input of `Trace`, i.e., the tracing key tk, is public then we say that the scheme supports public traceability.

We require that `Setup`, `Enc` and `Dec` run in polynomial time, and that with overwhelming probability over the randomness used by the algorithms, we have

$$\forall M \in \mathcal{P}, \forall i \le N : \texttt{Dec}(sk_i, \texttt{Enc}(pk, M)) = M,$$

where pk and the sk_i's are sampled from `Setup`. We also require the encryption scheme to be IND-CPA.

Algorithm `Trace` aims at deterring coalitions of malicious users (traitors) from building an unauthorized decryption device. It takes as input tk and has access to a decryption device $\mathcal{D}$. `Trace` aims at disclosing the identity of at least one user that participated in building $\mathcal{D}$.

We consider the minimal black-box access model [11]. In this model, the tracing authority has access to an oracle $\mathcal{O}^{\mathcal{D}}$ that itself internally uses $\mathcal{D}$. Oracle $\mathcal{O}^{\mathcal{D}}$ behaves as follows: It takes as input any pair $(C, M) \in \mathcal{C} \times \mathcal{P}$ and returns 1 if $\mathcal{D}(C) = M$ and 0 otherwise; the oracle only tells whether the decoder decrypts C to M or not. We assume that if M is sampled from $U(\mathcal{P})$ and C is the output of algorithm `Enc` given pk and M as inputs, then the decryption device decrypts correctly with probability significantly more than $1/|\mathcal{P}|$:

$$\Pr_{\substack{M \hookleftarrow U(\mathcal{P}) \\ C \hookleftarrow \texttt{Enc}(M)}} \left[\mathcal{O}^{\mathcal{D}}(C, M) = 1\right] \ge \frac{1}{|\mathcal{P}|} + \frac{1}{\lambda^c},$$

for some constant $c > 0$. This assumption is justified by the fact that otherwise the decryption device is not very useful. Alternatively, we may force the correct decryption probability to be non-negligibly close to 1, by using an all-or-nothing transform (see [33]). We also assume that the decoder $\mathcal{D}$ is stateless/resettable, i.e., it cannot see and adapt to it being tested and replies independently to successive queries. Handling stateful pirate boxes has been investigated in [31, 30].

In our scheme, algorithm `Trace` will only be a confirmation algorithm. It takes as input a set of (suspect) users $(\mathcal{U}_{i_j})_j$ of cardinality $k \le t$, and must satisfy the following two properties:

- Confirmation. If the traitors are all in the set of suspects $(\mathcal{U}_{i_j})_{j\le k}$, then it returns "User $\mathcal{U}_{i_{j_0}}$ is guilty" for some $j_0 \le k$;
- Soundness. If it returns "User $\mathcal{U}_{i_{j_0}}$ is guilty" for some $j_0 \le k$, then user $\mathcal{U}_{i_{j_0}}$ should indeed be a traitor.

The confirmation algorithm should run in polynomial-time. It may be converted into a (costly) full-fledge tracing algorithm by calling it on all subsets of users of cardinality t.

2.2 Euclidean lattices and discrete Gaussian distributions

A lattice is a set of the form $\{\sum_{i\le n} x_i \boldsymbol{b}_i : x_i \in \mathbb{Z}\}$ where the $\boldsymbol{b}_i$'s are linearly independent vectors in $\mathbb{R}^m$. In this situation, the $\boldsymbol{b}_i$'s are said to form a basis of the n-dimensional lattice. The nth minimum $\lambda_n(L)$ of an n-dimensional lattice L is defined as the smallest r such that the n-dimensional closed hyperball of radius r centered in $\mathbf{0}$ contains n linearly independent vectors of L. The smoothing parameter of L is defined as $\eta_\varepsilon(L) = \min\{r > 0 : \rho_{1/r}(\widehat{L} \setminus \mathbf{0}) \le \varepsilon\}$ for any $\varepsilon \in (0,1)$, where $\widehat{L} = \{\boldsymbol{c} \in \mathrm{Span}(L) : \boldsymbol{c}^t \cdot L \subseteq \mathbb{Z}\}$ is the dual lattice of L. It was proved in [43, Le. 3.3] that $\eta_\varepsilon(L) \le \sqrt{\ln(2n(1+1/\varepsilon))/\pi} \cdot \lambda_n(L)$ for all $\varepsilon \in (0,1)$ and n-dimensional lattices L.

For a lattice $L \subseteq \mathbb{R}^m$, a vector $\boldsymbol{c} \in \mathbb{R}^m$ and an invertible $S \in \mathbb{R}^{m\times m}$, we define the Gaussian distribution of parameters L, $\boldsymbol{c}$ and S by $D_{L,S,\boldsymbol{c}}(\boldsymbol{b}) \sim \rho_{S,\boldsymbol{c}}(\boldsymbol{b}) = \exp(-\pi\|S^{-1}(\boldsymbol{b}-\boldsymbol{c})\|^2)$ for all $\boldsymbol{b} \in L$. When $S = \sigma \cdot I_m$, we simply write $D_{L,\sigma,\boldsymbol{c}}$. Note that $D_{L,S,\boldsymbol{c}} = S^t \cdot D_{S^{-t}L,1,S^{-t}\boldsymbol{c}}$. Sometimes, for convenience, we use the notation $D_{L+\boldsymbol{c},S}$ as a shorthand for $\boldsymbol{c} + D_{L,S,-\boldsymbol{c}}$. Gentry et al. [27] showed that Klein's algorithm [34] can be used to sample from $D_{L,S,\boldsymbol{c}}$. This discrete Gaussian sampler was later refined in [18].

Lemma 1 ([18, Le. 2.3]). *There exists a ppt algorithm that, given a basis $(\boldsymbol{b}_i)_i$ of an n-dimensional lattice L, $\boldsymbol{c} \in \mathrm{Span}(L)$ and $S \in \mathbb{R}^{m\times m}$ invertible satisfying $\sqrt{\ln(2n+4)/\pi} \cdot \max_i \|S^{-t}\boldsymbol{b}_i\| \le 1$, returns a sample from $D_{L,S,\boldsymbol{c}}$.*

The following basic results on lattice Gaussians are usually stated for full-rank lattices. As we consider lattices that are not full-rank, we adapt them. The proofs can be modified readily to handle this more general setup, by relying on an isometry from $\mathrm{Span}(L)$ to $\mathbb{R}^n$ with $n = \dim L$.

Lemma 2 (Adapted from [5, Le. 3]). *For any n-dimensional lattice $L \subseteq \mathbb{R}^m$, $\boldsymbol{c} \in \mathrm{Span}(L)$ and $S \in \mathbb{R}^{m\times m}$ invertible satisfying $\sigma_m(S) \ge \eta_\varepsilon(L)$ with $\varepsilon \in (0,1/2)$, we have $\Pr_{\boldsymbol{b}\hookleftarrow D_{L,S,\boldsymbol{c}}}[\|\boldsymbol{b}-\boldsymbol{c}\| \ge \sigma_1(S)\cdot\sqrt{n}] \le 2^{-n+2}$.*

Lemma 3 (Adapted from [43, Le. 4.4]). *For any lattice $L \subseteq \mathbb{R}^m$, $\boldsymbol{c} \in \mathrm{Span}(L)$ and $S \in \mathbb{R}^{m\times m}$ invertible satisfying $\sigma_m(S) \ge \eta_\varepsilon(L)$ with $\varepsilon \in (0,1/2)$, we have $\rho_{S,\boldsymbol{c}}(L) \in (\frac{1-\varepsilon}{1+\varepsilon}, 1) \cdot \rho_S(L)$.*

Lemma 4 (Special case of [50, Th. 3.1]). *Let $S_1, S_2 \in \mathbb{R}^{m\times m}$ invertible, $\boldsymbol{c} \in \mathbb{R}^m$, and $\Lambda_1, \Lambda_2 \subseteq \mathbb{R}^m$ be full-rank lattices. Assume that $1 \ge \eta_\varepsilon(S_1^{-1}\Lambda_1)$ and $1 \ge \eta_\varepsilon(\sqrt{(S_1S_1^t)^{-1} + (S_2S_2^t)^{-1}} \cdot \Lambda_2)$ for some $\varepsilon \in (0,1/2)$. If $\boldsymbol{x}_2 \hookleftarrow D_{\Lambda_2,S_2,\mathbf{0}}$ and $\boldsymbol{x}_1 \hookleftarrow D_{\Lambda_1,S_1,\boldsymbol{c}-\boldsymbol{x}_2}$, then the residual distribution of $\boldsymbol{x}_1$ is within statistical distance 8ε of $D_{\Lambda_1,S,\boldsymbol{c}}$, with $S = \sqrt{S_1S_1^t + S_2S_2^t}$.*

Lemma 5 (Adapted from [12, Le. 4.12]). *Let $S_1, S_2 \in \mathbb{R}^{m\times m}$ invertible, $\boldsymbol{c}_1, \boldsymbol{c}_2 \in \mathbb{R}^m$, and $\Lambda_1, \Lambda_2 \subseteq \mathbb{R}^m$ be lattices. Assume that $1 \ge \eta_\varepsilon(\sqrt{(S_1S_1^t)^{-1} + (S_2S_2^t)^{-1}} \cdot \Lambda_1 \cap \Lambda_2)$ for some $\varepsilon \in (0,1/2)$. If $\boldsymbol{x}_1 \hookleftarrow D_{\Lambda_1,S_1,\boldsymbol{c}_1}$ and $\boldsymbol{x}_2 \hookleftarrow D_{\Lambda_2,S_2,\boldsymbol{c}_2}$, then the residual distribution of $\boldsymbol{x}_1 + \boldsymbol{x}_2$ is within statistical distance 4ε of $D_{\Lambda_1+\Lambda_2,S,\boldsymbol{c}_1+\boldsymbol{c}_2}$, where $S = \sqrt{S_1S_1^t + S_2S_2^t}$.*

Lemma 6 ([5, Le. 8]). *Let $n \ge 1$, $m \ge 2n$, and $\sigma \ge C \cdot \sqrt{n}$ for some absolute constant C. Let $X \hookleftarrow D_{\mathbb{Z},\sigma}^{m\times n}$. Then, except with probability $2^{-\Omega(m)}$, we have $\sigma_n(X) \ge \Omega(\sigma\sqrt{m})$.*

Lemma 7 ([2, Th. 5.1]). *Let $n \geq 100$, $\varepsilon \in (0, 1/1000)$, $\sigma \geq 9\sqrt{\ln(2n(1+1/\varepsilon))/\pi}$ and $m \geq 30n\log(\sigma n)$. Let $\boldsymbol{c} \in \mathbb{R}^m$ and $X \hookleftarrow D_{\mathbb{Z},\sigma}^{m\times n}$. Let $S \in \mathbb{R}^{m\times m}$ with $\sigma_m(S) \geq 10n\sigma\log^{3/2}(nm\sigma/\varepsilon)$. Then, with probability $\geq 1-2^{-n}$ over the choice of X, we have $X^t \cdot \mathbb{Z}^m = \mathbb{Z}^n$ and $\Delta(X^t \cdot D_{\mathbb{Z}^m,S,\boldsymbol{c}}, D_{\mathbb{Z}^n,S',X^t\boldsymbol{c}}) \leq 2\varepsilon$ with $S' = \sqrt{(SX)(SX)^t}$.*

We consider the following extension of Lemma 7, in which we force the first row of X to be the all-1 vector. This modification is important for our traitor tracing scheme, as having the first entry of every column of X equal to 1 is what enables correct decryption. Note that Lemma 16 and Theorem 17 also hold without this restriction on the first row of X (as can be seen by substituting Lemma 7 for Lemma 8 in their proofs).

Lemma 8. *Let $n \geq 100$, $\varepsilon \in (0, 1/1000)$, $\sigma \geq C \cdot \max(\sqrt{n}, \sqrt{\ln(n/\varepsilon)})$ for some constant $C > 0$ and $m \geq 30n\log(\sigma n) + 1$. Let $\boldsymbol{c} \in \mathbb{R}^m$ and $X \in \mathbb{Z}^{m\times n}$ with all entries of its first row equal to 1 and all remaining entries sampled from $D_{\mathbb{Z},\sigma}$. Let $S \in \mathbb{R}^{m\times m}$ invertible of the form $S = \left(\begin{array}{c|c}\sigma_1 & 0 \\ \hline 0 & T\end{array}\right)$, with $\sigma_m(S) \geq 10n\sigma\log^{3/2}(nm\sigma/\varepsilon)$. Then, with probability $\geq 1-2^{-n}$ over the choice of X, we have $X^t \cdot \mathbb{Z}^m = \mathbb{Z}^n$ and $\Delta(X^t \cdot D_{\mathbb{Z}^m,S,\boldsymbol{c}}, D_{\mathbb{Z}^n,S',X^t\boldsymbol{c}}) \leq 2\varepsilon$ with $S' = \sqrt{(SX)(SX)^t}$.*

Proof. Write $\boldsymbol{c} = (c_1 \| \boldsymbol{d})$ with $\boldsymbol{d} \in \mathbb{R}^{m-1}$ and $X = (\mathbf{1}^t \| Y)$ where $\mathbf{1}$ denotes the all-1 vector and $Y \hookleftarrow D_{\mathbb{Z},\sigma}^{(m-1)\times n}$. By Lemma 7, we have $Y^t \cdot \mathbb{Z}^{m-1} = \mathbb{Z}^n$ and $\Delta(Y^t \cdot D_{\mathbb{Z}^{m-1},T,\boldsymbol{d}}, D_{\mathbb{Z}^n,T',Y^t\boldsymbol{d}}) \leq 2\varepsilon$ with $T' \in \mathbb{Z}^{n\times n}$ invertible so that $T' = \sqrt{(TY)^t(TY)}$. This implies that $X^t \cdot \mathbb{Z}^m = \mathbb{Z}^n$. Now, we have $X^t \cdot D_{\mathbb{Z}^m,S,\boldsymbol{c}} = \mathbf{1} \cdot D_{\mathbb{Z},\sigma_1,c_1} + Y^t \cdot D_{\mathbb{Z}^{m-1},T,\boldsymbol{d}}$. The latter is within statistical distance 2ε from $\mathbf{1} \cdot D_{\mathbb{Z},\sigma_1,c_1} + D_{\mathbb{Z}^n,T',Y^t\boldsymbol{d}}$. The result follows from Lemma 5, using $\Lambda_2 = \mathbb{Z}^n, \Lambda_1 = \mathbf{1} \cdot \mathbb{Z}$ and $S_1 = \frac{\sigma_1}{\sqrt{n}} I_n$. The hypothesis $1 \geq \eta_\varepsilon(\sqrt{(S_1S_1^t)^{-1} + (S_2S_2^t)^{-1}} \cdot \Lambda_1 \cap \Lambda_2)$ of Lemma 5 is implied by the assumption on $\sigma_m(S)$ and the fact thats $\sigma_n(TY) \geq \sigma_m(S)\sigma_n(Y)$ and $\sigma_n(Y) \geq \Omega(\sqrt{m}\sigma)$ (by Lemma 6). ☐

We extensively use q-ary lattices. The q-ary lattice associated to $A \in \mathbb{Z}_q^{m\times n}$ is defined as $\Lambda^\perp(A) = \{\boldsymbol{x} \in \mathbb{Z}^m : \boldsymbol{x}^t \cdot A = \mathbf{0} \bmod q\}$. It has dimension m, and a basis can be computed in polynomial-time from A. For $\boldsymbol{u} \in \mathbb{Z}_q^m$, we define $\Lambda_{\boldsymbol{u}}^\perp(A)$ as the coset $\{\boldsymbol{x} \in \mathbb{Z}^m : \boldsymbol{x}^t \cdot A = \boldsymbol{u}^t \bmod q\}$ of $\Lambda^\perp(A)$.

2.3 Random lattices

We consider the following random lattices, called q-ary Ajtai lattices. They are obtained by sampling $A \hookleftarrow U(\mathbb{Z}_q^{m\times n})$ and considering $\Lambda^\perp(A)$. The following lemma provides a probabilistic bound on the smoothing parameter of $\Lambda^\perp(A)$.

Lemma 9 (Adapted from [27, Le. 5.3]). *Let q be prime and m, n integers with $m \geq 2n$ and $\varepsilon > 0$, then $\eta_\varepsilon(\Lambda^\perp(A)) \leq 4q^{\frac{n}{m}}\sqrt{\log(2m(1+1/\varepsilon))/\pi}$, for all except a fraction $2^{-\Omega(n)}$ of $A \in \mathbb{Z}_q^{m\times n}$.*

It is possible to efficiently sample a close to uniform A along with a short basis of $\Lambda^\perp(A)$ (see [7, 8, 50, 42]).

Lemma 10 (Adapted from [8, Th. 3.1]). *There exists a ppt algorithm that given $n, m, q \geq 2$ as inputs samples two matrices $A \in \mathbb{Z}_q^{m\times n}$ and $T \in \mathbb{Z}^{m\times m}$ such that: the distribution of A is within statistical distance $2^{-\Omega(n)}$ from $U(\mathbb{Z}_q^{m\times n})$; the rows of T form a basis of $\Lambda^\perp(A)$; each row of T has norm $\leq 3mq^{n/m}$.*

For $A \in \mathbb{Z}_q^{m \times n}$, $S \in \mathbb{R}^{m \times m}$ invertible, $\boldsymbol{c} \in \mathbb{R}^m$ and $\boldsymbol{u} \in \mathbb{Z}_q^n$, we define the distribution $D_{\Lambda_{\boldsymbol{u}}^{\perp}(A),S,\boldsymbol{c}}$ as $\bar{\boldsymbol{c}} + D_{\Lambda^{\perp}(A),S,-\bar{\boldsymbol{c}}+\boldsymbol{c}}$, where $\bar{\boldsymbol{c}}$ is any vector of $\mathbb{Z}^m$ such that $\bar{\boldsymbol{c}}^t \cdot A = \boldsymbol{u}^t \bmod q$. A sample $\boldsymbol{x}$ from $D_{\Lambda_{\boldsymbol{u}}^{\perp}(A),S}$ can be obtained using the GPV algorithm along with the short basis of $\Lambda^{\perp}(A)$ provided by Lemma 10. Boneh and Freeman [12] showed how to efficiently obtain the residual distribution of $(A, \boldsymbol{x})$ without relying on Lemma 10.

Theorem 11 (Adapted from [12, Th. 4.3]). *Let $n, m, q \geq 2$, $k \geq 0$ and $S \in \mathbb{R}^{m \times m}$ be such that $m \geq 2n$, q is prime with $q > \sigma_1(S) \cdot \sqrt{2\log(4m)}$, and $\sigma_m(S) = q^{\frac{n}{m}} \cdot \max(\Omega(\sqrt{n \log m}), 2\sigma_1(S)^{\frac{k}{m}})$. Let $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k \in \mathbb{Z}_q^n$ and $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k \in \mathbb{R}^m$ be arbitrary. Then the residual distributions of the tuple $(A, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)$ obtained with the following two experiments are within statistical distance $2^{-\Omega(n)}$.*

$$\begin{aligned}
\mathtt{Exp}_0 : &\quad A \hookleftarrow U(\mathbb{Z}_q^{m \times n}); \qquad \forall i \leq k : \boldsymbol{x}_i \hookleftarrow D_{\Lambda_{\boldsymbol{u}_i}^{\perp}(A),S,\boldsymbol{c}_i} \,. \\
\mathtt{Exp}_1 : &\, \forall i \leq k : \boldsymbol{x}_i \hookleftarrow D_{\mathbb{Z}^m,S,\boldsymbol{c}_i};\, A \hookleftarrow U\left(\mathbb{Z}_q^{m \times n} | \forall i \leq k : \boldsymbol{x}_i^t \cdot A = \boldsymbol{u}_i^t \bmod q\right).
\end{aligned}$$

This statement generalizes [12, Th. 4.3] in three ways. First, the latter corresponds to the special case corresponding to taking all the $\boldsymbol{u}_i$'s and $\boldsymbol{c}_i$'s equal to $\boldsymbol{0}$. This generalization does not add any extra complication in the proof of [12, Th. 4.3], but is important for our constructions. Second, the condition on m is less restrictive (the corresponding assumption in [12, Th. 4.3] is that $m \geq \max(2n \log q, 2k)$). To allow for such small values of m, we refine the bound on the smoothing parameter of the $\Lambda^{\perp}(A)$ lattice (namely, we use Lemma 9). Third, we allow for a non-spherical Gaussian distribution, as we are able to adapt the Micciancio-Peikert trapdoor technique to kernel lattices only with non-spherical Gaussian distribution for the trapdoor vectors.

We also use the following result on the probability of the Gaussian vectors $\boldsymbol{x}_i$ from Theorem 11 being linearly independent over $\mathbb{Z}_q$.

Lemma 12 (Adapted from [12, Le. 4.5]). *With the notations and assumptions of Theorem 11, the k vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ sampled in $\mathtt{Exp}_0$ and $\mathtt{Exp}_1$ are linearly independent over $\mathbb{Z}_q$, except with probability $2^{-\Omega(n)}$.*

2.4 Rényi divergence

We use Rényi Divergence (RD) in our analysis, relying on techniques developed in [41, 38, 9]. For any two probability distributions P and Q such that the support of P is a subset of the support of Q over a countable domain X, we define the RD (of order 2) by $R(P\|Q) = \sum_{x \in X} \frac{P(x)^2}{Q(x)}$, with the convention that the fraction is zero when both the numerator and denominator are zero. We recall that the RD between two offset discrete Gaussians is bounded as follows.

Lemma 13 ([38, Le. 4.2]). *For any n-dimensional lattice $L \subseteq \mathbb{R}^n$ and invertible matrix S, set $P = D_{L,S,\boldsymbol{w}}$ and $Q = D_{L,S,\boldsymbol{z}}$ for some fixed $\boldsymbol{w}, \boldsymbol{z} \in \mathbb{R}^n$. If $\boldsymbol{w}, \boldsymbol{z} \in L$, let $\varepsilon = 0$. Otherwise, fix $\varepsilon \in (0,1)$ and assume that $\sigma_n(S) \geq \eta_\varepsilon(L)$. Then $R(P\|Q) \leq \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^2 \cdot \exp\left(2\pi \|\boldsymbol{w} - \boldsymbol{z}\|^2 / \sigma_n(S)^2\right)$.*

We use this bound and the fact that the RD between the parameter distributions of two distinguishing problems can be used to relate their hardness, if they satisfy a certain public samplability property.

Lemma 14 ([9]). *Let Φ, Φ' denote two distributions with* $\mathrm{Supp}(\Phi) \subseteq \mathrm{Supp}(\Phi')$*, and $D_0(r)$ and $D_1(r)$ denote two distributions determined by some parameter $r \in \mathrm{Supp}(\Phi')$. Let P, P' be two decision problems defined as follows:*

- *P: Assess whether input x is sampled from distribution X_0 or X_1, where*

$$X_0 = \{x : r \hookleftarrow \Phi, x \hookleftarrow D_0(r)\},\ X_1 = \{x : r \hookleftarrow \Phi, x \hookleftarrow D_1(r)\}.$$

- *P': Assess whether input x is sampled from distribution X'_0 or X'_1, where*

$$X'_0 = \{x : r \hookleftarrow \Phi', x \hookleftarrow D_0(r)\},\ X'_1 = \{x : r \hookleftarrow \Phi', x \hookleftarrow D_1(r)\}.$$

Assume that $D_0(\cdot)$ and $D_1(\cdot)$ have the following public samplability *property: there exists a sampling algorithm* S *with run-time T_S such that for all r, b, given any sample x from $D_b(r)$ we have:*

- $\mathsf{S}(0, x)$ *outputs a sample distributed as $D_0(r)$ over the randomness of* S.
- $\mathsf{S}(1, x)$ *outputs a sample distributed as $D_1(r)$ over the randomness of* S.

If there exists a T-time distinguisher $\mathcal{A}$ for problem P with advantage ε, then we can construct a distinguisher $\mathcal{A}'$ for problem P' with run-time and advantage respectively bounded from above and below by:

$$O\left(\frac{1}{\varepsilon^2}\log\left(\frac{R(\Phi\|\Phi')}{\varepsilon}\right)\cdot(T_S + T)\right) \quad \textit{and} \quad \frac{\varepsilon^3}{8R(\Phi\|\Phi')}.$$

2.5 Learning with errors

Let $\boldsymbol{s} \in \mathbb{Z}_q^n$ and $\alpha > 0$. We define the distribution $A_{\boldsymbol{s},\alpha}$ as follows: Take $\boldsymbol{a} \hookleftarrow U(\mathbb{Z}_q^n)$ and $e \hookleftarrow \nu_\alpha$, and return $(\boldsymbol{a}, \frac{1}{q}\langle \boldsymbol{a}, \boldsymbol{s}\rangle + e) \in \mathbb{Z}_q^n \times \mathbb{T}$. The *Learning With Errors problem* LWE_α, introduced by Regev in [56, 57], consists in assessing whether an oracle produces samples from $U(\mathbb{Z}_q^n \times \mathbb{T})$ or $A_{\boldsymbol{s},\alpha}$ for some constant $\boldsymbol{s} \hookleftarrow U(\mathbb{Z}_q^n)$. Regev [57] showed that for $q \leq \mathcal{P}oly(n)$ prime and $\alpha \in (\frac{\sqrt{n}}{2q}, 1)$, LWE is (quantumly) not easier than standard worst-case lattice problems in dimension n with approximation factors $\mathcal{P}oly(n)/\alpha$. This hardness proof was partly dequantized in [49, 18], and the requirements that q should be prime and $\mathcal{P}oly(n)$ were waived.

In this work, we consider a variant LWE where the number of oracle samples that the distinguisher requests is a priori bounded. If m denotes that bound, then we will refer to this restriction as $\mathrm{LWE}_{\alpha,m}$. In this situation, the hardness assumption can be restated in terms of linear algebra over $\mathbb{Z}_q$: Given $A \hookleftarrow U(\mathbb{Z}_q^{m\times n})$, the goal is to distinguish between the distributions (over $\mathbb{T}^m$)

$$\frac{1}{q}U\left(\mathrm{Im}(A)\right) + \nu_\alpha^m \ \text{ and } \ \frac{1}{q}U\left(\mathbb{Z}_q^m\right) + \nu_\alpha^m.$$

Under the assumption that $\alpha q \geq \Omega(\sqrt{n})$, the right hand side distribution is indeed within statistical distance $2^{-\Omega(n)}$ to $U(\mathbb{T}^m)$ (see, e.g., [43, Le. 4.1]). The hardness assumption states that by adding to them a small Gaussian noise, the linear spaces $\mathrm{Im}(A)$ and $\mathbb{Z}_q^m$ become computationally indistinguishable. This rephrasing in terms of linear algebra is helpful in the security proof of the traitor tracing scheme. Note that by a standard hybrid argument, distinguishing between the two distributions given one sample from either, and distinguishing between them given Q samples (from the same distribution), are computationally equivalent problems, up to a loss of a factor Q in the distinguishing advantage.

Finally, we will also use a variant of LWE where the noise distribution ν_α is replaced by $D_{q^{-1}\mathbb{Z},\alpha}$, and where $U(\mathbb{T})$ is replaced by $U(\mathbb{T}_q)$ with $\mathbb{T}_q$ being $q^{-1}\mathbb{Z}$ with addition mod 1. This variant, denoted by LWE$'$, was proved in [50] to be no easier than standard LWE (up to a constant factor increase in α).

3 New lattice tools

The security of our constructions relies on the hardness of a new variant of LWE, which may be seen as the dual of the k-SIS problem from [12].

Definition 15. Let $k \leq m$, $S \in \mathbb{R}^{m\times m}$ invertible and $C = (\boldsymbol{c}_1\|\cdots\|\boldsymbol{c}_k) \in \mathbb{R}^{m\times k}$. The (k, S, C)-LWE$_{\alpha,m}$ problem (or (k, S)-LWE if $C = 0$) is as follows: Given $A \hookleftarrow U(\mathbb{Z}_q^{m\times n})$, $\boldsymbol{u} \hookleftarrow U(\mathbb{Z}_q^n)$ and $\boldsymbol{x}_i \hookleftarrow D_{\Lambda^{\perp}_{-\boldsymbol{u}}(A),S,\boldsymbol{c}_i}$ for $i \leq k$, the goal is to distinguish between the distributions (over $\mathbb{T}^{m+1}$)

$$\frac{1}{q} \cdot U\Big(\mathrm{Im}\Big(\frac{\boldsymbol{u}^t}{A}\Big)\Big) + \nu_\alpha^{m+1} \quad \text{and} \quad \frac{1}{q} \cdot U\Big(\mathrm{Span}_{i\leq k}\Big(\frac{1}{\boldsymbol{x}_i}\Big)^{\perp}\Big) + \nu_\alpha^{m+1}.$$

The classical LWE problem consists in distinguishing the left distribution from uniform, without the hint vectors $\boldsymbol{x}_i^+ = (1\|\boldsymbol{x}_i)$. These hint vectors correspond to the secret keys obtained by the malicious coalition in the traitor tracing scheme. Once these hint vectors are revealed, it becomes easy to distinguish the left distribution from the uniform distribution: take one of the vectors $\boldsymbol{x}_i^+$, get a challenge sample $\boldsymbol{y}$ and compute $\langle \boldsymbol{x}_i^+, \boldsymbol{y}\rangle \in \mathbb{T}$; if $\boldsymbol{y}$ is a sample from the left distribution, then the centered residue is expected to be of size $\approx \alpha \cdot (\sqrt{m}\sigma_1(S) + \|\boldsymbol{c}_i\|)$, which is $\ll 1$ for standard parameter settings; on the other hand, if $\boldsymbol{y}$ is sampled from the uniform distribution, then $\langle \boldsymbol{x}^+, \boldsymbol{y}\rangle$ should be uniform. The definition of (k, S)-LWE handles this issue by replacing $U(\mathbb{Z}_q^{m+1})$ by $U(\mathrm{Span}_{i\leq k}(\boldsymbol{x}_i^+)^{\perp})$.

Sampling $\boldsymbol{x}_i^+$ from $D_{\Lambda^{\perp}((\boldsymbol{u}^t\|A)),S,\boldsymbol{c}_i}$ may seem more natural than imposing that the first coordinate of each $\boldsymbol{x}_i^+$ is 1. Looking ahead, this constraint will prove convenient to ensure correctness of our cryptographic primitives. Theorem 18 below and its proof can be readily adapted to this hint distribution. They may also be adapted to improve the SIS to k-SIS reduction from [12]. Setting $C = 0$ is also more natural, but for technical reasons, our reduction from LWE to (k, S, C)-LWE works with unit vectors $\boldsymbol{c}_i$. However, we show that for small $\|\boldsymbol{c}_i\|$, there exist polynomial time reductions between (k, S, C)-LWE and (k, S)-LWE.

In the proof of the hardness of the (k, S)-LWE problem, we rely on an adaptation of the Micciancio-Peikert trapdoor construction [42] for q-ary lattices, to sampling a Gaussian $X \in \mathbb{Z}^{m\times n}$ along with a short basis for the lattice $\ker(X) = \{\boldsymbol{b} \in \mathbb{Z}^m : \boldsymbol{b}^t X = \boldsymbol{0}\}$. This is the purpose of the next subsection.

3.1 Sampling a Gaussian X with a small basis of $\ker(X)$

The Micciancio-Peikert construction [42] relies on a *leftover hash lemma* stating that with overwhelming probability over $A \hookleftarrow U(\mathbb{Z}_q^{m\times n})$ and for a sufficiently large σ, the distribution of $A^t \cdot D_{\mathbb{Z}^m,\sigma}$ mod q is statistically close to $U(\mathbb{Z}_q^n)$. We use a similar result over the integers, starting from a Gaussian $X \in \mathbb{Z}^{m\times n}$ instead of a uniform $A \in \mathbb{Z}_q^{m\times n}$. The proof of the following lemma relies on [2], which improves over a similar result from [5]. The result would be neater with $\sigma_2 = \sigma_1$, but, unfortunately, we do not know how to achieve it. The impact of this drawback on our results and constructions is mostly cosmetic.

Lemma 16. *Let $m > n \geq 100$ and $\sigma_1, \sigma_2 > 0$ satisfying $\sigma_1 \geq \Omega(\sqrt{mn \log m})$, $m \geq \Omega(n \log(\sigma_1 n))$ and $\sigma_2 \geq \Omega(n^{5/2}\sqrt{m}\sigma_1^2 \log^{3/2}(m\sigma_1))$. Let $X = (\mathbf{1}^t \| Y) \in \mathbb{Z}^{m \times n}$ with all entries of its first row equal to 1 and all remaining entries sampled from $D_{\mathbb{Z},\sigma_1}$. There exists a ppt algorithm that takes $n, m, \sigma_1, \sigma_2, X$ and $\boldsymbol{c} \in \mathbb{Z}^n$ as inputs and returns $\boldsymbol{x} \in \mathbb{Z}^n, \boldsymbol{r} \in \mathbb{Z}^m$ such that $\boldsymbol{x} = \boldsymbol{c} + X^t \boldsymbol{r}$ with $\|\boldsymbol{r}\| \leq O(\sigma_2/\sigma_1)$, with probability $1 - 2^{-\Omega(n)}$, and*

$$\Delta((Y, \boldsymbol{x}), D_{\mathbb{Z},\sigma_1}^{(m-1)\times n} \times D_{\mathbb{Z}^n,\sigma_2,\boldsymbol{c}}) \leq 2^{-\Omega(n)}.$$

Proof. We apply Lemma 8 with S of the correct shape, invertible and chosen so that $(SX)^t(SX) = \sigma_2^2 I_n$ for some $\sigma_2 > \sigma_1$, thus obtaining an unskewed Gaussian distribution $D_{\mathbb{Z}^n,\sigma_2,\boldsymbol{c}}$. The scaling σ_2 is chosen sufficiently large so that the assumptions of Lemmas 6 and 8 hold.

We first sample X as in the result statement, using Lemma 1, and write $X = (\mathbf{1}^t \| Y)$. By Lemma 8 (that we use with $\varepsilon = 2^{-n}$), its row $\mathbb{Z}$-span is $\mathbb{Z}^n$ with probability $\geq 1 - 2^{-n}$: we now assume that we are in this situation. Then we sample $\boldsymbol{r}$ from $D_{\mathbb{Z}^m,S}$, using Lemma 1 again, for some invertible matrix $S \in \mathbb{R}^{m \times m}$ chosen as described below. Finally, we set $\boldsymbol{x} = \boldsymbol{c} + X^t \cdot \boldsymbol{r}$. If the assumptions of Lemma 8 are satisfied, we know that, except with probability $\leq 2^{-n}$ over X, the distribution of $\boldsymbol{x}$ is, conditioned on X, within statistical distance 2ε of $D_{\mathbb{Z}^n,\sigma_2,\boldsymbol{c}}$.

We build $S = \left(\begin{array}{c|c} \sigma_2/\sqrt{2} & 0 \\ \hline 0 & T \end{array}\right)$ using the singular value decomposition $Y = U_Y \cdot \mathrm{Diag}((\sigma_i(Y))_{i \leq n}) \cdot V_Y$, where $U_Y \in \mathbb{R}^{(m-1)\times n}$ and $V_Y \in \mathbb{R}^{n \times n}$ are orthogonal matrices. We define $T = U_T \cdot \mathrm{Diag}((t_i)_{i<m}) \cdot V_T$ as follows: we set $U_T^t = \begin{bmatrix} V_Y & \mathbf{0} \\ \mathbf{0} & I_{m-1-n} \end{bmatrix}$ and $V_T^t = [U_Y | U_Y^\perp]$, where $U_Y^\perp$ is an orthonormal basis for the orthogonal of $U_Y \cdot \mathbb{R}^n$; we also set $t_i = \sigma_2/(\sqrt{2} \cdot \sigma_i(Y))$ for $i \leq n$ and $t_i = \sigma_n(T)$ for $i > n$. This leads to $(TY)^t(TY) = (\sigma_2^2/2) \cdot I_n$. We thus have $(SX)^t(SX) = \sigma_2^2 \cdot I_n$, as required.

To check that the assumptions of Lemma 8 are satisfied, note that the smallest singular value of S is $\sigma_m(S) = \sigma_2/(\sqrt{2} \cdot \sigma_1(Y))$. Hence the assumption $\sigma_m(S) \geq 10n\sigma_1 \log^{3/2}(nm\sigma_1/\varepsilon)$ is satisfied if $\sigma_2/\sqrt{2} \geq \sigma_1(Y) \cdot 10n\sigma_1 \log^{3/2}(nm\sigma_1/\varepsilon)$. The latter holds by the choice of σ_2, using the fact that $\sigma_1(Y) \leq \|Y\| \leq \sqrt{m} \cdot \sigma_1$. The second inequality holds with probability $\geq 1 - n2^{-m+2}$, using the union bound and Lemma 2.

Finally, the bound on $\|\boldsymbol{r}\|$ follows from Lemma 2 and the facts that $\sigma_1(S) = \sigma_2/(\sqrt{2} \cdot \sigma_n(Y))$ and $\sigma_n(Y) \geq \Omega(\sigma_1 \sqrt{m})$ except with probability $2^{-\Omega(m)}$, by Lemma 6. □

We now adapt the trapdoor construction from [42] to kernel lattices.

Theorem 17. *Let $n, m_1, \sigma_1, \sigma_2$ be as above, and $m_2 \geq m_1$ bounded as $n^{O(1)}$. There exists a ppt algorithm that given n, m_1, m_2 (in unary), σ_1 and σ_2, returns $X_1 \in \mathbb{Z}^{m_1 \times n}, X_2 \in \mathbb{Z}^{m_2 \times n}$, and $U \in \mathbb{Z}^{m \times m}$ with $m = m_1 + m_2$, such that:*

- *the distribution of (X_1, X_2) is within statistical distance $2^{-\Omega(n)}$ of the distribution $D_{\mathbb{Z},\sigma_1}^{m_1 \times n} \times (D_{\mathbb{Z}^{m_2},\sigma_2,\boldsymbol{\delta}_1} \times \cdots \times D_{\mathbb{Z}^{m_2},\sigma_2,\boldsymbol{\delta}_n})$ conditioned on all entries of the first row of X_1 being equal to 1, and where $\boldsymbol{\delta}_i$ denotes the ith canonical unit vector in $\mathbb{Z}^{m_2}$ whose ith coordinate is 1 and whose remaining coordinates are 0,*
- *we have $|\det U| = 1$ and $U \cdot X = (I_n \| 0)$ with $X = (X_1 \| X_2)$,*
- *every row of U has norm $\leq O(\sqrt{nm_1}\sigma_2)$ with probability $\geq 1 - 2^{-\Omega(n)}$.*

The second statement implies that the last $m - n$ rows of U form a basis of the random lattice $\ker(X)$.

Proof. We first sample X_1 as in Lemma 16, using the GPV algorithm. We run m_2 times the algorithm from Lemma 16, on the input $n, m_1, \sigma_1, \sigma_2, X_1$ and $\boldsymbol{c}$ running through the columns of $C = [I_n | 0_{n\times(m_2-n)}]$. This gives $X_2 \in \mathbb{Z}^{m_2\times n}$ and $R \in \mathbb{Z}^{m_1\times m_2}$ such that $X_2^t = [I_n|\mathbf{0}_{n\times(m_2-n)}] + X_1^t \cdot R$. One can then see that $U \cdot X = [I_n \| \mathbf{0}]$, where

$$U = \left[\begin{array}{c|c} \mathbf{0} & I_{m_2} \\ \hline I_{m_1} & -(X_1|\mathbf{0}) \end{array}\right] \cdot \left[\begin{array}{c|c} I_{m_1} & \mathbf{0} \\ \hline -R^t & I_{m_2} \end{array}\right] = \left[\begin{array}{c|c} -R^t & I_{m_2} \\ \hline I_{m_1} + (X_1|\mathbf{0})R^t & -(X_1|\mathbf{0}) \end{array}\right], X = \left[\begin{array}{c} X_1 \\ \hline X_2 \end{array}\right].$$

The result then follows from Gaussian tail bounds (to bound the norms of the rows of X_1) and elementary computations. □

3.2 Hardness of *k*-LWE

The following result shows that this LWE variant, with S a specific diagonal matrix, is no easier than LWE.

Theorem 18. *Let m, q, σ, σ', k be such that $\sigma \geq \Omega(\max(m\sqrt{\log m}, \sigma'^{k/(m+k)}))$, $\sigma' \geq \Omega(m^3\sigma^2 \log^{3/2}(m\sigma))$, $q \geq \Omega(\sigma'\sqrt{\log m})$ is prime, and $m \geq \Omega(n \log q)$. Then there exists a probabilistic polynomial-time reduction from $\mathrm{LWE}_{\alpha,m+1}$ in dimension n to (k, S)-$\mathrm{LWE}_{\alpha',m+1+k}$ in dimension n, with $\alpha' = \Omega(m^{3/2}\sigma'\alpha)$ and $S = \left[\begin{array}{c|c} \sigma \cdot I_{m+k} & 0 \\ \hline 0 & \sigma' \cdot I_n \end{array}\right]$. More concretely, using a (k, S)-$\mathrm{LWE}_{\alpha',m+k}$ algorithm with run-time T and advantage ε, the reduction gives an $\mathrm{LWE}_{\alpha,m+1}$ algorithm with run-time $T' = O(T \cdot \mathcal{P}oly(m) \cdot (\varepsilon')^{-2} \log((\varepsilon')^{-1}))$ and advantage $\varepsilon' = \Omega((\varepsilon')^3)$, where $\varepsilon' = \varepsilon - 2^{-\Omega(n)}$.*

Example parameters are $k = m/10$, $\sigma = \widetilde{\Theta}(n)$, $\sigma' = \widetilde{\Theta}(n^5)$, $q = \widetilde{\Theta}(n^5)$ and $m = \Theta(n \log n)$.

The reduction takes an LWE instance and extends it to a related k-LWE instance for which the additional hint vectors $(\boldsymbol{x}_i)_{i\leq k}$ are known. The major difficulty in this extension is to restrain the noise increase, as a function of k.

The existing approach for this reduction (that we improve below) is the technique used in the SIS to k-SIS reduction from [12]. In the latter approach, the hint vectors are chosen independently from a small discrete Gaussian distribution, and then the LWE matrix $A \in \mathbb{Z}_q^{m\times n}$ is extended to a taller matrix A' under the constraint that the hint vectors are in the q-ary lattice $\Lambda^{\perp}(A') = \{\boldsymbol{b} : \boldsymbol{b}^t A' = \mathbf{0} \bmod q\}$. Unfortunately, with this approach, the transformation from an LWE sample with respect to A, to a k-LWE sample with respect to A', involves a multiplication by the cofactor matrix $\det(G) \cdot G^{-1}$ over $\mathbb{Z}$ of a $k \times k$ full-rank submatrix G of the hint vectors matrix. Although the entries of G are small, the entries of its cofactor matrix are almost as large as $\det G$, which is exponential in k. This leads to an "exponential noise blowup," restraining the applicability range to $k \leq \widetilde{O}(1)$ if one wants to rely on the hardness of LWE with noise rate $1/\alpha \leq \mathcal{P}oly(n)$ (otherwise, LWE is not exponentially hard to solve).

To restrain the noise increase for large k, we use a matrix pair $(X, U) \in \mathbb{Z}^{k\times(m+k)} \times \mathbb{Z}^{(m+k)\times(m+k)}$ from Theorem 17. We write $X = (\mathbf{1}^t \| Y)$ and let $V \in \mathbb{Z}^{m\times(m+k)}$ be the matrix made of the bottom m rows of U (it is a basis of $\ker(X)$). At a high level, our reduction consists in mapping the LWE instance $(A, \boldsymbol{b})$ to the k-LWE instance $(V^t A, Y^t, V^t \boldsymbol{b})$. Note that $X^t(V^t A) = 0 \bmod q$, which means that the columns of Y are indeed hint vectors. In the proof of correctness of the reduction, we will argue that the pair $(V^t A, Y)$ is almost correctly distributed. The distribution imperfection lies in the fact that some rows of Y have non-zero centers, because of Theorem 17.

In the proof, we will argue that if an algorithm can efficiently solve k-LWE with non-zero-centered hints, it can be used to efficiently solve k-LWE with zero-centered hints. Another difficulty is that the noise component of $V^t\boldsymbol{b}$ is skewed. We will unskew it by adding to it some extra Gaussian noise, with a carefully designed covariance matrix.

Proof. Without loss of generality, we assume that $k = \Omega(n)$ throughout the proof: the result with large values of k implies the result for smaller values of k, as we can always disregard hints. We also assume that $k = O(m)$, as otherwise the conditions on σ and σ' cannot be satisfied simultaneously.

For a technical reason related to the non-zero centers $\boldsymbol{\delta}_i$ in the distribution of the hint vectors produced by Theorem 17, we decompose our reduction from $\mathrm{LWE}_{\alpha,m+1}$ to (k,S)-LWE into two subreductions. The first subreduction (outlined above) reduces $\mathrm{LWE}_{\alpha,m+1}$ to (k,S,C)-$\mathrm{LWE}_{\alpha',m+1+k}$, where the ith column of C is the unit vector $\boldsymbol{c}_i = (0^m\|\boldsymbol{\delta}_i) \in \mathbb{R}^{m+k}$ for $i = 1,\ldots,k$. The second subreduction reduces (k,S,C)-$\mathrm{LWE}_{\alpha',m+1+k}$ to (k,S)-$\mathrm{LWE}_{\alpha',m+1+k}$. Both reductions preserve the LWE dimension n. We first describe and analyze the first subreduction, and then explain the second subreduction.

Description of the first subreduction. Let $(A^+,\boldsymbol{b})$ with $A^+ = (\boldsymbol{u}^t\|A)$ denote the given $\mathrm{LWE}_{\alpha,m+1}$ input instance, where $A^+ \hookleftarrow U(\mathbb{Z}_q^{(m+1)\times n})$, and $\boldsymbol{b} \in \mathbb{T}^{m+1}$ comes from either the "LWE distribution" $\frac{1}{q}U\left(\mathrm{Im}(A^+)\right) + \nu_\alpha^{m+1}$ or the "Uniform distribution" $\frac{1}{q}U\left(\mathbb{Z}_q^{m+1}\right) + \nu_\alpha^{m+1}$. The reduction maps $(A^+,\boldsymbol{b})$ to $(A',\boldsymbol{u}',X,\boldsymbol{b}')$ with $A' \in \mathbb{Z}_q^{(m+k)\times n}$ and $\boldsymbol{u}' \in \mathbb{Z}_q^n$ independent and uniform, $X \in \mathbb{Z}^{k\times(m+k)}$ with its ith row $\boldsymbol{x}_i$ independently sampled from $D_{\Lambda^\perp_{-\boldsymbol{u}'}(A'),S,\boldsymbol{c}_i}$ for $i \le k$, and $\boldsymbol{b}' \in \mathbb{T}^{m+1+k}$ coming from either the "k-LWE distribution" $\frac{1}{q}U\left(\mathrm{Im}(A'^+)\right) + \nu_\alpha^{m+1+k}$ if $\boldsymbol{b}$ is from the "LWE distribution," or the "k-Uniform distribution" $\frac{1}{q}U\left(\mathrm{Span}_{i\le k}(\boldsymbol{x}_i^+)^\perp\right)$ if $\boldsymbol{b}$ is from the "Uniform distribution." Here $A'^+ = (\boldsymbol{u}'^t\|A')$, and $\boldsymbol{x}_i^+$ denotes the vector $(1\|\boldsymbol{x}_i)$ for $i \le k$. The reduction is as follows.

1. Sample $(X_1,X_2,U) \in \mathbb{Z}^{(m+1)\times k} \times \mathbb{Z}^{k\times k} \times \mathbb{Z}^{(m+1+k)\times(m+1+k)}$ as in Theorem 17 (with parameters n, m_1, m_2, σ_1 and σ_2 respectively set to $k, m+1, k, \sigma$ and σ'). Define $\boldsymbol{x}_i^+$ as the ith column of $(X_1\|X_2)$ for $i \le k$ (recall that all the entries of the first row of X_1 are equal to 1). Let $V \in \mathbb{Z}^{(m+1)\times(m+1+k)}$ be the matrix consisting of the bottom $m+1$ rows of U. Let $X \in \mathbb{Z}^{k\times(m+k)}$ be the matrix whose ith row is $\boldsymbol{x}_i$ for all $i \le k$.
2. Compute $\Sigma = \alpha' \cdot I_{m+1+k} - V^t \cdot V$ and $\sqrt{\Sigma}$ such that $\sqrt{\Sigma}^t \cdot \sqrt{\Sigma} = \Sigma$; if Σ is not positive definite, abort.
3. Compute $A'^+ = (V^t \cdot A^+)$ and $\boldsymbol{b}' = V^t\boldsymbol{b} + \sqrt{\Sigma}\boldsymbol{e}'$, with $\boldsymbol{e}' \hookleftarrow \nu_1^{m+1+k}$. Let $(\boldsymbol{u}')^t$ be the top row of A'^+.
4. Return $(A',\boldsymbol{u}',X,\boldsymbol{b}')$.

Step 1 aims at building a transformation matrix V^t that sends A^+ to A'^+. Two properties are required from this transformation. First, it must be a linear map with small coefficients, so that when we map the LWE right hand side to the k-LWE right hand side, the noise component does not blow up. Second, it must contain some vectors $(1\|\boldsymbol{x}_i)$ in its (left) kernel, with $\boldsymbol{x}_i$ normally distributed. These vectors are to be used as k-LWE hints. For this, we use Theorem 17. This ensures that the $\boldsymbol{x}_i$'s are (almost) distributed as independent Gaussian samples from $D_{\mathbb{Z}^m,\sigma} \times D_{\mathbb{Z}^k,\sigma'}$, and that the matrix V is integral with small coefficients. We have, for all $i \le k$:

$$[1|\boldsymbol{x}_i^t] \cdot A'^+ = [1|\boldsymbol{x}_i^t] \cdot V^t \cdot A^+ = \boldsymbol{0} \bmod q.$$

This means each $\boldsymbol{x}_i$ belongs to $\Lambda^{\perp}_{-\boldsymbol{u}}(A')$.

The fact that U is unimodular ensures that the columns of A'^{+} are uniformly distributed in the subspace orthogonal to the hint vectors $\boldsymbol{x}_i^{+}$ modulo q. This shows that the reduction maps "Uniform" to "k-Uniform".

It remains to see how to map "LWE" to "k-LWE." The main problem, when multiplying $\boldsymbol{b}$ by V^t, is that the LWE noise gets skewed. If its covariance matrix was of the form $\alpha^2 \cdot I_{m+1}$, then it becomes $\alpha^2 V \cdot V^t$. To compensate for that, in Step 3, we add to $V^t \cdot \boldsymbol{b}$ an independent Gaussian noise with well-chosen covariance matrix, designed at Step 2. At Step 2, we set α' large enough to ensure that the symmetric matrix $\alpha' \cdot I_{m+1+k} - V^t \cdot V$ is positive definite. This noise unskewing technique was adapted to discrete Gaussians in [50].

Analysis of the first subreduction. All steps of the reduction can be implemented in polynomial time. Its correctness follows from the following three lemmas.

Lemma 19. *The tuple $(A', \boldsymbol{u}', X)$ is within statistical distance $2^{-\Omega(n)}$ of the distribution in which $A' \in \mathbb{Z}_q^{(m+k)\times n}$ and $\boldsymbol{u}' \in \mathbb{Z}_q^n$ are independent and uniform, and the rows of $X \in \mathbb{Z}^{k\times(m+k)}$ are from $D_{\Lambda^{\perp}_{-\boldsymbol{u}'}(A'),S,\boldsymbol{c}_i}$, where $\boldsymbol{c}_i = (0^m \| \boldsymbol{\delta}_i) \in \mathbb{R}^{m+k}$ and $\boldsymbol{\delta}_i$ denotes the ith canonical unit vector in $\mathbb{Z}^k$ for $i = 1, \ldots, k$.*

Proof. Let D_0 denote the desired distribution for $(A', \boldsymbol{u}', X)$. We first apply Theorem 11 (with the theorem parameters $m, n, k, \sigma_1(S), \sigma_m(S)$ having the values $m+k$, n, k, σ' and σ, respectively) to show that D_0 is within statistical distance $2^{-\Omega(n)}$ of the distribution D_1 on tuples $(A', \boldsymbol{u}', X)$ defined as follows: $\boldsymbol{u}' \in \mathbb{Z}_q^n$ is sampled uniformly, $X \in \mathbb{Z}^{k\times(m+k)}$ has its ith row $\boldsymbol{x}_i$ independently sampled from $D_{\mathbb{Z}^{m+k},S,\boldsymbol{c}_i}$, and $A' \in \mathbb{Z}_q^{(m+k)\times n}$ is sampled uniformly from the set of solutions to $\boldsymbol{x}_i^t \cdot A' = -\boldsymbol{u}'^t \bmod q$. Indeed, the assumptions of the theorem are satisfied by our choice of parameters.

Let D_2 denote the distribution of $(A', \boldsymbol{u}', X)$ in the reduction. We show below that $\Delta(D_1, D_2) \leq 2^{-\Omega(n)}$, which completes the proof.

By Theorem 17, we have that, in distribution D_2, the distribution of X is within statistical distance $2^{-\Omega(n)}$ of the distribution of X in D_1. Now, we consider the distribution of A'^{+}. By construction, we have that $A'^{+} = V^t A^{+}$. This implies that $\boldsymbol{x}_i^t \cdot A' = -\boldsymbol{u}'^t \bmod q$ for all i. As X has full rank modulo q (by Lemma 12), the number of solutions A' to this equation is exactly q^{mn}. Further, as V can be extended into a unimodular matrix, multiplication by V^t is an injective map, and the cardinality of $V^t\mathbb{Z}_q^{m\times n}$ is exactly q^{mn}. This implies that $V^t\mathbb{Z}_q^{m\times n}$ is exactly the set of solutions to the equation, and that $A' = V^t A$ is uniformly distributed in that set. □

Next, we assume that (A'^{+}, X) is fixed and consider the distribution of $\boldsymbol{b}'$ in the two cases of the distribution of $\boldsymbol{b}$. First we consider the "LWE" to "k-LWE" distribution mapping.

Lemma 20. *The following holds with probability $1 - 2^{-\Omega(n)}$ over the choice of X. If $\boldsymbol{b} \in \mathbb{T}^{m+1}$ is sampled from $\frac{1}{q}U(\mathrm{Im}A^{+}) + \nu_\alpha^{m+1}$, then $\boldsymbol{b}' \in \mathbb{T}^{m+1+k}$ is within statistical distance $2^{-\Omega(n)}$ of $\frac{1}{q}U\left(\mathrm{Im}A'^{+}\right) + \nu_{\alpha'}^{m+1+k}$.*

Proof. We have $\boldsymbol{b} = \frac{1}{q}A^{+} \cdot \boldsymbol{s} + \boldsymbol{e} \in \mathbb{T}^{m+1}$ with $\boldsymbol{e}$ sampled from ν_α^{m+1} and $\boldsymbol{s}$ from $U(\mathbb{Z}_q^n)$, so

$$\boldsymbol{b}' = V^t \cdot \boldsymbol{b} + \sqrt{\Sigma} \cdot \boldsymbol{e}'$$

$$= \frac{1}{q} V^t A^+ \cdot \boldsymbol{s} + V^t \cdot \boldsymbol{e} + \sqrt{\Sigma} \cdot \boldsymbol{e}'$$
$$= \frac{1}{q} A'^+ \cdot \boldsymbol{s} + V^t \cdot \boldsymbol{e} + \sqrt{\Sigma} \cdot \boldsymbol{e}'.$$

Now, since $\boldsymbol{s}$ is uniform, we have $\frac{1}{q} A'^+ \cdot \boldsymbol{s}$ is uniformly distributed in $\mathrm{Im}(A'^+)$. Moreover, the vector $V^t \cdot \boldsymbol{e}$ is normally distributed with covariance matrix $\alpha^2 \cdot V V^t$, while $\sqrt{\Sigma} \boldsymbol{e}'$ is independent and normally distributed with covariance matrix $\Sigma = \alpha'^2 I_{m+1+k} - \alpha^2 V^t V$ (we show below that Σ is indeed a valid covariance matrix, i.e., is positive definite, so that $\sqrt{\Sigma}$ exists, except with probability $2^{-\Omega(n)}$). Therefore, the vector $V^t \cdot \boldsymbol{e} + \sqrt{\Sigma} \boldsymbol{e}'$ has distribution $\nu_{\alpha'}^{m+1+k}$, as required.

It remains to show that $\Sigma = \alpha'^2 I_{m+1+k} - \alpha^2 V^t V$ is a positive definite matrix, with overwhelming probability over the choice of X. By definition, the singular values of Σ are of the form $\alpha'^2 - \alpha^2 \sigma_i(V)^2$, where the $\sigma_i(V)$'s are the singular values of V. It therefore suffices to show that $\alpha'^2 > \alpha^2 \sigma_1(V)^2$, where $\sigma_1(V)$ is the largest singular value of V. We have $\sigma_1(V) \leq \sqrt{m+1} \|V\|$ (by Schwarz's inequality). By Theorem 17, we have that $\|V\| \leq O(\sqrt{km}\sigma')$. It follows that $\sigma_1(V) = O(mk^{1/2}\sigma')$, and hence the assumption that $\alpha' = \Omega(m^{3/2}\sigma'\alpha)$ allows us to complete the proof. ☐

Finally, we consider the "Uniform" to "k-Uniform" distribution mapping.

Lemma 21. *The following holds with probability* $1 - 2^{-\Omega(n)}$ *over the choice of* X*. If* $\boldsymbol{b}$ *is sampled from* $\frac{1}{q} U\left(\mathbb{Z}_q^{m+1}\right) + \nu_\alpha^{m+1}$*, then* $\boldsymbol{b}'$ *is within statistical distance* $2^{-\Omega(n)}$ *of* $\frac{1}{q} U\left(\mathrm{Span}_{i \leq k}(\boldsymbol{x}_i^+)^\perp\right) + \nu_{\alpha'}^{m+1+k}$.

Proof. We have $\boldsymbol{b} = \frac{1}{q}\boldsymbol{y} + \boldsymbol{e} \in \mathbb{T}^{m+1}$ with $\boldsymbol{e}$ sampled from ν_α^{m+1} and $\boldsymbol{y}$ from $U(\mathbb{Z}_q^{m+1})$, so

$$\boldsymbol{b}' = \frac{1}{q} V^t \cdot \boldsymbol{y} + V^t \cdot \boldsymbol{e} + \sqrt{\Sigma} \cdot \boldsymbol{e}'.$$

Now, since $\boldsymbol{y}$ is uniform, we have that $\frac{1}{q} V^t \boldsymbol{y}$ is uniform in $\mathrm{Im} V^t = \mathrm{Span}_{i \leq k}(\boldsymbol{x}_i^+)^\perp$, as explained in the proof of Lemma 19. As shown in Lemma 20, we also have that the noise term $V^t \cdot \boldsymbol{e} + \sqrt{\Sigma} \boldsymbol{e}'$ is within statistical distance $2^{-\Omega(n)}$ of the distribution ν_α^{m+1+k}, as required. ☐

Overall, we have described a reduction that maps the "LWE distribution" to the "k-LWE distribution," and the "Uniform distribution" to the "k-Uniform distribution," up to statistical distance $2^{-\Omega(n)}$.

Second subreduction. It remains to reduce the (k, S, C)-LWE with non-zero centers for the hint distribution, to (k, S)-LWE with zero-centered hints. For this, we use Lemma 14 to obtain the following.

Lemma 22. *Let* $m' = m + k$ *and assume that* $\sigma_{m'}(S) \geq \omega(\sqrt{n})$*. If there exists a distinguisher against* (k, S)-$\mathrm{LWE}_{\alpha', m'}$ *(in dimension* n*) with run-time* T *and advantage* ε*, then there exists a distinguisher against* (k, S, C)-$\mathrm{LWE}_{\alpha', m'}$ *with run-time* $T' = O(\mathcal{P}oly(m') \cdot (\varepsilon')^{-2} \cdot T \cdot \log \frac{R}{\varepsilon'})$ *and advantage* $\varepsilon' = \Omega((\varepsilon')^3 / R - 2^{-\Omega(n)})$*, where* $\varepsilon' = \varepsilon - 2^{-\Omega(n)}$ *and* $R = \exp(O(k \cdot (2^{-n} + \|C\|^2 / \sigma_{m'}(S)^2)))$.

Proof. Consider the following sequence of games.

Let Game_0 denote the original (k, S)-LWE game, in which the distinguisher $\mathcal{B}$ receives an instance of the form $(r, \boldsymbol{y})$, where $r = (A, \boldsymbol{u}, \{\boldsymbol{x}_i\}_{i \leq k})$ with $A \hookleftarrow U(\mathbb{Z}_q^{m' \times n})$,

$\boldsymbol{u} \hookleftarrow U(\mathbb{Z}_q^n)$ and $\boldsymbol{x}_i \hookleftarrow D_{\Lambda^\perp_{-\boldsymbol{u}}(A),S,\mathbf{0}}$ for $i \leq k$, and $\boldsymbol{y} \in \mathbb{T}^{m'+1}$ is a sample from either the distribution

$$D_0(r) = \frac{1}{q} \cdot U\Big(\mathrm{Im}\Big(\frac{\boldsymbol{u}^t}{A}\Big)\Big) + \nu_\alpha^{m'+1}$$

or the distribution

$$D_1(r) = \frac{1}{q} \cdot U\Big(\mathrm{Span}_{i \leq k}\Big(\frac{1}{\boldsymbol{x}_i}\Big)^\perp\Big) + \nu_\alpha^{m'+1}.$$

Let $\varepsilon_0(\mathcal{B}) = \varepsilon$ denote the advantage of $\mathcal{B}$ in distinguishing between these distributions in Game_0. Similarly, in the following, we let $\varepsilon_i(\mathcal{B})$ denote the corresponding attacker advantage in Game_i.

Let Game_1 denote a modification of Game_0 in which we change the distribution of A by rejection sampling as follows: we sample A uniformly from $\mathbb{Z}_q^{m' \times n}$, but reject and resample A if $\eta_{2^{-n}}(A) > 4q^{n/m'}\sqrt{\log(2m'(1+2^n))/\pi} = O(\sqrt{n})$. By Lemma 9, the rejection probability is $2^{-\Omega(n)}$, and therefore, the distinguishing advantage $\varepsilon_1(\mathcal{B})$ satisfies $\varepsilon_1(\mathcal{B}) \geq \varepsilon_0(\mathcal{B}) - 2^{-\Omega(n)}$.

Let Game_2 denote a modification of Game_1 in which we change the distribution of the hint $\boldsymbol{x}_i$'s in r from the zero-centered distribution $D_{\Lambda^\perp_{-\boldsymbol{u}}(A),S,\mathbf{0}}$ of (k,S)-LWE to the non-zero centered distribution $D_{\Lambda^\perp_{-\boldsymbol{u}}(A),S,\boldsymbol{c}_i}$ of (k,S,C)-LWE. We observe that, since given $r = (A, \boldsymbol{u}, \{\boldsymbol{x}_i\}_{i \leq k})$, one can efficiently sample a vector $\boldsymbol{y}$ from either distribution $D_0(r)$ or $D_1(r)$, the (k,S,C)-LWE problem has the *public samplability* property needed to apply Lemma 14. It follows that there exists a distinguisher $\mathcal{B}'$ in Game_2 with run-time $T' = O(\mathcal{P}oly(m') \cdot \varepsilon_1(\mathcal{B})^{-2} \cdot T)$ and advantage $\varepsilon_2(\mathcal{B}') \geq \Omega((\varepsilon_1(\mathcal{B}) - O(2^{-n}))^3/R)$, where $R = R(\Phi_1 \| \Phi_2)$ denotes the RD between the distributions Φ_1 and Φ_2 of r in Game_1 and Game_2, respectively. Since the $\boldsymbol{x}_i$'s are independent, and conditioning on $\boldsymbol{u}$ and A, we have, from the multiplicative property of the RD, that

$$\begin{aligned} R &\leq \max_{\boldsymbol{u} \in \mathbb{Z}_q^{n'}} \prod_{i \leq k} R\left(D_{\Lambda^\perp_{-\boldsymbol{u}}(A),S,\mathbf{0}} \| D_{\Lambda^\perp_{-\boldsymbol{u}}(A),S,\boldsymbol{c}_i}\right) \\ &\leq \max_{\bar{\boldsymbol{c}} \in \mathbb{R}^{m'}} \prod_{i \leq k} R\left(D_{\Lambda^\perp(A),S,\bar{\boldsymbol{c}}} \| D_{\Lambda^\perp(A),S,\bar{\boldsymbol{c}}+\boldsymbol{c}_i}\right). \end{aligned}$$

The latter can be bounded from above by applying Lemma 13. The smoothing condition of the lemma holds since $\sigma_{m'}(S) \geq \omega(\sqrt{n})$, so we have by the rejection step of the previous game that $\sigma_{m'}(S) \geq \eta_{2^{-n}}(A)$. This leads to

$$R \leq \prod_{i \leq k} \exp(2^{-n+3} + 2\pi\|\boldsymbol{c}_i\|^2/\sigma_{m'}(S)^2) \leq \exp(k \cdot (2^{-n+3} + 2\pi\|C\|^2/\sigma_{m'}(S)^2)).$$

Finally, let Game_3 denote a modification of Game_2, in which we undo the rejection sampling of A introduced in Game_1, sampling it uniformly instead. By the same argument as in the change from Game_0 to Game_1, the advantage of $\mathcal{B}'$ in Game_3 satisfies $\varepsilon_3(\mathcal{B}') \geq \varepsilon_2(\mathcal{B}') - 2^{-\Omega(n)}$. Note that the instance distribution in Game_3 is identical to that of the (k,S,C)-LWE game, so $\mathcal{B}'$ has advantage $\varepsilon_3(\mathcal{B}')$ against (k,S,C)-LWE, as required. □

In our application of Lemma 22, the (k,S,C)-LWE problem resulting from the first subreduction has $\|C\| = 1$, and $\sigma_{m'}(S) = \sigma$, so that $R = \exp(O(k \cdot (2^{-n} + 1/\sigma^2))) = O(1)$ using $\sigma = \Omega(n)$ and $k \leq n$. This shows that the second subreduction is probabilistic polynomial time. □

Our technique can be applied to improve the Boneh-Freeman reduction from SIS to k-SIS, from an exponential loss in k to a polynomial loss in k. In fact, we map A to $A' = V^t A$ in the same way (except that we do not use and add $\boldsymbol{u}$ on top of the matrix A) and use X as the k-SIS hints for the new matrix A'. Then, whenever the adversary can output a short vector $\boldsymbol{x}$ that is orthogonal to A', we can also output a short vector $V\boldsymbol{x}$ which is orthogonal to A. As $\boldsymbol{x}$ is linearly independent from the rows of X and V is a basis of the lattice $\ker X$, we obtain that $V\boldsymbol{x} \neq \mathbf{0}$. Also, the vector $V\boldsymbol{x}$ is small, as both V and $\boldsymbol{x}$ have small entries. RD may also be used to reduce k-SIS with non-zero-centered hints (with small centers) to k-SIS with zero-centered hints.

4 A lattice-based public-key traitor tracing scheme

In this section, we describe and analyze our basic traitor tracing scheme. First, we give the underlying multi-user public-key encryption scheme. We then explain how to implement black-box confirmation tracing.

4.1 A multi-user encryption scheme

The scheme is designed for a given security parameter n, a number of users N and a maximum malicious coalition size t. It then involves several parameters q, m, α, S. These are set so that the scheme is correct (decryption works properly on honestly generated ciphertexts) and secure (semantically secure encryption and possibility to trace members of malicious coalitions). In particular, we set $S = \mathrm{Diag}(\sigma, \ldots, \sigma, \sigma', \ldots, \sigma') \in \mathbb{R}^{m \times m}$ where $\sigma' > \sigma$ and their respective numbers of iterations are set so that (t, S)-$\mathrm{LWE}_{\alpha,m+1}$ is hard to solve.

`Setup`. The trusted authority generates a master key pair using the algorithm from Lemma 10. Let $(A, T) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}^{m \times m}$ be the output. We additionally sample $\boldsymbol{u}$ uniformly in $\mathbb{Z}_q^n$. Matrix T will be part of the tracing key tk, whereas the public key is $pk = A^+$, with $A^+ = (\boldsymbol{u}^t \| A)$.

Each user $\mathcal{U}_i$ for $i \leq N$ obtains a secret key sk_i from the trusted authority, as follows. The authority executes the GPV algorithm using the basis of $\Lambda^\perp(A)$ consisting of the rows of T, and the standard deviation matrix S. The authority obtains a sample $\boldsymbol{x}_i$ from $D_{\Lambda^\perp_{-\boldsymbol{u}}(A),S}$. The standard deviations $\sigma' > \sigma$ may be chosen as small as $3mq^{n/m}\sqrt{(2m+4)/\pi}$. The user secret key is $\boldsymbol{x}_i^+ = (1 \| \boldsymbol{x}_i) \in \mathbb{Z}^{m+1}$. Using the Gaussian tail bound and the union bound, we have $\|\boldsymbol{x}_i\| \leq \sqrt{m}\sigma'$ for all $i \leq N$, with probability $\geq 1 - N \cdot 2^{-\Omega(m)}$.

The tracing key tk consists of the matrix T and all pairs $(\mathcal{U}_i, sk_i)$.

`Encrypt`. The encryption algorithm is exactly the 1-bit encryption scheme from [27, Se. 7.1], which we recall, for readability.[1] The plaintext and ciphertext domains are $\mathcal{P} = \{0,1\}$ and $\mathcal{C} = \mathbb{Z}_q^{m+1}$ respectively, and:

$$\mathtt{Enc} : M \mapsto \left[\frac{\boldsymbol{u}^t}{A}\right] \cdot \boldsymbol{s} + \boldsymbol{e} + \left[\frac{M \cdot \lfloor q/2 \rfloor}{\mathbf{0}}\right], \quad \text{where } \boldsymbol{s} \hookleftarrow U(\mathbb{Z}_q^n) \text{ and } \boldsymbol{e} \hookleftarrow \lfloor \nu_{\alpha q} \rceil^{m+1}.$$

As explained in [27], this scheme is semantically secure under chosen plaintext attacks (IND-CPA), under the assumption that $\mathrm{LWE}_{\alpha,m+1}$ is hard to solve.

[1] As usual, the encryption algorithm may be used to encapsulate session keys which are then fed into an efficient data encapsulation mechanism to encrypt the data.

`Decrypt`. To decrypt a ciphertext $\boldsymbol{c} \in \mathbb{Z}_q^{m+1}$, user $\mathcal{U}_i$ uses its secret key $\boldsymbol{x}_i^+$ and evaluates the following function `Dec` from $\mathbb{Z}_q^{m+1}$ to $\{0,1\}$: Map $\boldsymbol{c}$ to 0 if $\langle \boldsymbol{x}_i^+, \boldsymbol{c} \rangle \bmod q$ is closer to 0 than $\pm\lfloor q/2 \rfloor$.

If $\boldsymbol{c}$ is an honestly generated ciphertext of a plaintext $M \in \{0,1\}$, we have $\langle \boldsymbol{x}_i^+, \boldsymbol{c} \rangle = \langle \boldsymbol{x}_i^+, \boldsymbol{e} \rangle + M \cdot \lfloor q/2 \rfloor \bmod q$, where $\boldsymbol{e} \hookleftarrow \lfloor \nu_{\alpha q} \rceil^{m+1}$. It can be shown that the latter has magnitude $\leq 2\sqrt{m}\alpha q \|\boldsymbol{x}_i^+\|$ with probability $1 - 2^{-\Omega(n)}$ over the randomness of $\boldsymbol{e}$. This is $\leq 3m\alpha q\sigma'$ for all i, with probability $\geq 1 - N \cdot 2^{-\Omega(n)}$. To ensure the correctness of the scheme, it suffices to set $q \geq 4m\alpha q\sigma'$. Note that other constraints will be added to enable tracing.

Theorem 23. *Let m, n, q and N be integers such that q is prime and $N \leq 2^{o(n)}$. Let $\alpha, \sigma, \sigma' > 0$ such that $\sigma' \geq \sigma \geq \Omega(mq^{n/m}\sqrt{\log m})$ and $\alpha \leq 1/(4m\sigma')$. Then the scheme described above is IND-CPA under the assumption that $\mathrm{LWE}_{m+1,\alpha}$ is hard. Further, the decryption algorithm is correct:*

$$\forall M \in \{0,1\}, \forall i \leq N: \ \mathtt{Dec}\left(\mathtt{Enc}(M, pk), sk_i\right) = M$$

holds with probability $\geq 1 - 2^{-\Omega(n)}$ over the randomness used in `Setup` *and* `Enc`.

4.2 Tracing traitors

We now present a black-box confirmation algorithm `Trace`.[2] It is given access to an oracle $\mathcal{O}^{\mathcal{D}}$ that provides black-box access to a decryption device $\mathcal{D}$. It takes as inputs the tracing key $tk = (T, (\mathcal{U}_i, \boldsymbol{x}_i^+)_{i \leq N})$ and a set of suspect users $\{\mathcal{U}_{i_1}, \ldots, \mathcal{U}_{i_k}\}$ of cardinality $k \leq t$, where t is the a priori bound on any coalition size. Wlog, we may consider that $k = t$ and $i_j = j$ for all $j \leq k$.

Algorithm `Trace` gathers information about which keys have been used to build decoder $\mathcal{D}$, by feeding different carefully designed distributions to oracle $\mathcal{O}^{\mathcal{D}}$. We consider the following $t+1$ distributions $Tr_0, \ldots, Tr_t$ over $\mathcal{C} = \mathbb{Z}_q^{m+1}$:

$$Tr_i = U\left(\mathrm{Span}(\boldsymbol{x}_1^+, \ldots, \boldsymbol{x}_i^+)^\perp\right) + \lfloor \nu_{\alpha q} \rceil^{m+1}.$$

The first distribution Tr_0 is the uniform distribution, whereas the last distribution Tr_t is meant to be computationally indistinguishable from `Enc`(0). We define p_∞ as the probability $\Pr[\mathcal{O}^{\mathcal{D}}(\boldsymbol{c}, M) = 1]$ that the decoder can decrypt the ciphertexts, over the randomness of $M \hookleftarrow U(\{0,1\})$ and $\boldsymbol{c} \hookleftarrow \mathtt{Enc}(M)$. We define p_i as the probability the decoder decrypts the signals in Tr_i, for $i \in [0,t]$:

$$p_i \;=\; \Pr_{\substack{\boldsymbol{c} \hookleftarrow Tr_i \\ M \hookleftarrow U(\{0,1\})}} \left[\mathcal{O}^{\mathcal{D}}\left(\boldsymbol{c} + \begin{bmatrix} M \cdot \lfloor q/2 \rfloor \\ \boldsymbol{0} \end{bmatrix}, M\right) = 1\right].$$

A gap between p_{i-1} and p_i is meant to indicate that $\mathcal{U}_i$ is a traitor.

4.3 Confirmation and soundness of the proposed traitor tracing

We define the usefulness of the decoder as $\varepsilon := p_\infty - \frac{1}{|\mathcal{P}|} = p_\infty - \frac{1}{2}$. It can be estimated to within a factor 2 with probability $\geq 1 - 2^{-\Omega(n)}$ via the Chernoff bound.

We can now formally describe algorithm `Trace`. It proceeds in three steps, as follows.

[2] Note that in our context, minimal access is equivalent to standard access: since the plaintext domain is small, plaintext messages can be tested exhaustively.

1. It computes an estimate $\widetilde{\varepsilon}$ of the usefulness ε of the decoder to within a multiplicative factor of 2, which holds with probability $\geq 1-2^{-n}$. This can be obtained via Chernoff's bound, and costs $O(\varepsilon^{-2}n)$.
2. For i from 0 to t, algorithm `Trace` computes an approximation $\widetilde{p_i}$ of p_i to within an absolute error $\leq \frac{\widetilde{\varepsilon}}{16t}$, which holds with probability $\geq 1-2^{-n}$ (also using Chernoff's bound).
3. If $\widetilde{p_i} - \widetilde{p}_{i-1} > \frac{\widetilde{\varepsilon}}{8t}$ for some $i \leq t$, then `Trace` returns "User $\mathcal{U}_i$ is guilty." Otherwise, it returns "$\perp$."

Note that we are implicitly using the fact that $\mathcal{D}$ is stateless/resettable. Also, if ε is n^{-c} for some constant c, then `Trace` runs in polynomial time.

We start with the confirmation property.

Theorem 24. *Assume that decoder $\mathcal{D}$ was built using $\{sk_j\}_{j\leq k} \subseteq \{sk_i\}_{i\leq t}$. Under the assumption that (t,S)-LWE$_{\alpha,m+1}$ is hard, algorithm* `Trace` *returns "User $\mathcal{U}_i$ is guilty" for some $i \leq t$.*

Proof. Wlog we may assume that the traitors in the coalition know all the secret keys $sk_1,\ldots,sk_t$. The hardness of (t,S)-LWE$_{\alpha,m+1}$ implies that the distributions `Enc`(0) and Tr_t are computationally indistinguishable. As a consequence, we have that p_t is negligibly close to p_∞ (the rounding to nearest of the samples from $\nu_{\alpha q}$ can be performed directly on the challenge samples, obliviously to any secret data, as in the proof of semantic security of Section 4.1).

On the other hand, the acceptance probability p_0 is $\leq \frac{1}{2}$. As $p_t - p_0 > \frac{\varepsilon}{2}$ and $|\widetilde{p_i} - p_i| \leq \frac{\varepsilon}{8}$ for all i, we must have $\widetilde{p_t} - \widetilde{p_0} > \frac{\varepsilon}{4} \geq \frac{\widetilde{\varepsilon}}{8}$, with probability exponentially close to 1. As a consequence, there must exist $i \leq t$ such that $\widetilde{p_i} - \widetilde{p}_{i-1} > \frac{\widetilde{\varepsilon}}{8t}$, and algorithm `Trace` returns "User $\mathcal{U}_i$ is guilty." □

Proving the soundness property is more involved. We exploit the hardness of (t,S)-LWE and rely on Theorem 11 several times.

Theorem 25. *Assume that decoder $\mathcal{D}$ was built using $\{sk_j\}_{j\leq k}$. Under the parameter assumptions of Theorem 11 with (k,n) in Theorem 11 set to $(t+1, n+t+1)$, and the computational assumption that $(t+1,S)$-LWE$_{\alpha,m+1}$ is hard: if algorithm* `Trace` *returns "User $\mathcal{U}_{i_0}$ is guilty", then $i_0 \leq k$.*

Proof. Assume (by contradiction) that the traitors $\{\mathcal{U}_j\}_{j\leq k}$ with $k \leq t$ succeed in having `Trace` incriminate an innocent user $\mathcal{U}_{i_0}$ (with $i_0 > k$). We show that the algorithm $\mathcal{T}$ the traitors use to build the pirate decoder may be exploited for solving $(t+1,S)$-LWE$_{\alpha,m+1}$. First, note that algorithm $\mathcal{T}$ provides an algorithm $\mathcal{A}$ that wins the following game.

Game_0. The game consists of three steps, as follows:

- `Initialize`$_0$: Sample $A \hookleftarrow U(\mathbb{Z}_q^{m\times n})$, $\boldsymbol{u} \hookleftarrow U(\mathbb{Z}_q^n)$ and $\boldsymbol{x}_i \hookleftarrow D_{\Lambda^\perp_{-\boldsymbol{u}}(A),S}$ for $i \leq t+1$.
- `Input`$_0$: Send $A^+ = (\boldsymbol{u}^t \| A)$ and $(\boldsymbol{x}_i)_{i\leq t+1, i\neq i_0}$ to $\mathcal{A}$.
- `Challenge`$_0$: Sample $b \hookleftarrow U(\{0,1\})$. Send to $\mathcal{A}$ arbitrarily many samples from $U\left(\mathrm{Span}_{i\leq i_0-1+b}(\boldsymbol{x}_i^+)^\perp\right) + \lfloor \nu_{\alpha q} \rceil^{m+1}$.

We say that $\mathcal{A}$ wins Game_0 if it finds the value of b with non-negligible advantage.

Algorithm $\mathcal{A}$ can be obtained from algorithm $\mathcal{T}$ by sampling plaintext M uniformly in $\{0,1\}$, and giving $(\boldsymbol{c} + (M|\mathbf{0}^t)^t, M)$ as input to $\mathcal{O}^{\mathcal{D}}$, where $\boldsymbol{c}$ is any sample from

$\texttt{Challenge}_0$. We now introduce two variations of Game_0, which differ in the Initialize and Challenge steps.

Game_1. The game consists of three steps, as follows:

- $\texttt{Initialize}_1$: Sample $A \hookleftarrow U(\mathbb{Z}_q^{m\times n})$, $\boldsymbol{u} \hookleftarrow U(\mathbb{Z}_q^n)$, $\boldsymbol{x}_i \hookleftarrow D_{\Lambda^{\perp}_{-\boldsymbol{u}}(A),\sigma}$ for $i \leq t+1$, and $\boldsymbol{b}_j^+ \hookleftarrow U(\mathrm{Span}_{i<i_0}(\boldsymbol{x}_i^+)^{\perp})$ for $j \leq t - i_0 + 2$.
- $\texttt{Input}_1$: Send $A^+ = (\boldsymbol{u}^t \| A)$ and $(\boldsymbol{x}_i)_{i\leq t+1, i\neq i_0}$ to $\mathcal{A}$.
- $\texttt{Challenge}_1$: Sample $b \hookleftarrow U(\{0,1\})$. If $b=0$, then send to $\mathcal{A}$ arbitrarily many samples from $U\left(\mathrm{Span}_{i<i_0}(\boldsymbol{x}_i^+)^{\perp}\right) + \lfloor \nu_{\alpha q} \rceil^{m+1}$. If $b=1$, then send to $\mathcal{A}$ arbitrarily many samples from:

$$U\left(\mathrm{Im}\left[A^+|\boldsymbol{b}_1^+|\ldots|\boldsymbol{b}_{t-i_0+2}^+\right]\right) + \lfloor \nu_{\alpha q} \rceil^{m+1}.$$

As in Game_0, algorithm $\mathcal{A}$ wins Game_1 if it guesses b with non-negligible advantage.

Game_1' is as Game_1, except that if $b=0$ in the challenge step, then the samples sent to $\mathcal{A}$ are from the distribution $U\left(\mathrm{Span}_{i\leq i_0}(\boldsymbol{x}_i^+)^{\perp}\right) + \lfloor \nu_{\alpha q} \rceil^{m+1}$. (The $\boldsymbol{b}_j$'s are sampled from $U(\mathrm{Span}_{i<i_0}(\boldsymbol{x}_i^+)^{\perp})$ in both cases.)

Note that $\mathcal{A}$'s inputs in Game_0, Game_1 and Game_1' are identical (only the distributions of the Challenge steps vary). By the triangle inequality, if $\mathcal{A}$ wins Game_0 with some non-negligible advantage, then it may be used to win either Game_1 or Game_1' with non-negligible advantage. In our use of $\mathcal{A}$ to solve $(t+1,S)$-LWE, we may guess in which situation we are. We now consider the two situations separately.

First situation: Algorithm $\mathcal{A}$ wins Game_1 with non-negligible advantage. Then it may be used to solve $(t+1,S)$-LWE. Indeed, assume we have a $(t+1,S)$-LWE input $(A, \boldsymbol{u}, (\boldsymbol{x}_i)_{i\leq t+1})$, and that we aim at distinguishing between the following distributions over $\mathbb{Z}_q^{m+1}$:

$$U\left(\mathrm{Im}(A^+)\right) + \nu_{\alpha q}^{m+1} \quad \text{and} \quad U\left(\mathrm{Span}_{i\leq t+1}(\boldsymbol{x}_i^+)^{\perp}\right) + \nu_{\alpha q}^{m+1}.$$

To solve this problem instance, we sample $\boldsymbol{b}_j$ for $j \leq t - i_0 + 2$ as in $\texttt{Initialize}_1$. Then we add a uniform $\mathbb{Z}_q$-linear combination of the $\boldsymbol{b}_j$'s to the $(t+1,S)$-LWE input samples. Since $m \geq t+n$, these $(t-i_0+2)$ vectors are linearly independent and none of them belongs to $\mathrm{Span}_{i_0\leq i\leq t+1}(\boldsymbol{x}_i^+)^{\perp}$, with probability $\geq 1 - 2^{-\Omega(n)}$. In that case, the transformation maps $U\left(\mathrm{Span}_{i\leq t+1}(\boldsymbol{x}_i^+)^{\perp}\right) + \nu_{\alpha q}^{m+1}$ to $U\left(\mathrm{Span}_{i<i_0}(\boldsymbol{x}_i^+)^{\perp}\right) + \nu_{\alpha q}^{m+1}$, and maps $U(\mathrm{Im}(A^+)) + \nu_{\alpha q}^{m+1}$ to $U(\mathrm{Im}[A^+|\boldsymbol{b}_1^+|\ldots|\boldsymbol{b}_{t-i_0+2}^+]) + \nu_{\alpha q}^{m+1}$. We then round the samples to the nearest integer vectors, and Algorithm $\mathcal{A}$ distinguishes between the resulting distributions, and its output is forwarded as output to the initial $(t+1,S)$-LWE instance.

Second situation: Algorithm $\mathcal{A}$ wins Game_1' with non-negligible advantage. It seems quite similar to the first situation, but the following observation hints why its handling is somewhat more complex. In the first situation, the domains of the noiseless variants of the distributions to be distinguished are contained into one another: $\mathrm{Im}([A^+|\boldsymbol{b}_1|\ldots|\boldsymbol{b}_{t-i_0+2}]) \subseteq \mathrm{Span}_{i<i_0}(\boldsymbol{x}_i^+)^{\perp}$. In the second situation, no such inclusion holds. The purpose of the sequence of games below is to map Game_1' to recover such an inclusion setting.

Let us define Game_2 as being the same as Game_1', but with the following updated first step:

- $\texttt{Initialize}_2$: Sample $A \hookleftarrow U(\mathbb{Z}_q^{m\times n})$, $\boldsymbol{u} \hookleftarrow U(\mathbb{Z}_q^n)$, $\boldsymbol{b}_j \hookleftarrow U(\mathbb{Z}_q^m)$ and $v_j \hookleftarrow U(\mathbb{Z}_q)$ for $j \le t - i_0 + 2$, $\boldsymbol{x}_i \hookleftarrow D_{\Lambda^\perp_{-\boldsymbol{u}}(A),S}$ for $i \ge i_0$ and $\boldsymbol{x}_i \hookleftarrow D_{\Lambda^\perp_{-\boldsymbol{u}'}(A'),S}$ for $i < i_0$, with
$$A' = [A|\boldsymbol{b}_1|\ldots|\boldsymbol{b}_{t-i_0+2}] \quad \text{and} \quad \boldsymbol{u}' = (\boldsymbol{u}\|v_1\|\ldots\|v_{t-i_0+2}).$$

We show that the residual distributions at the end of $\texttt{Initialize}_1$ and $\texttt{Initialize}_2$ are essentially the same. For that, we use Theorem 11 twice. First, starting from $\texttt{Initialize}_1$, we swap the samplings of A and $\boldsymbol{u}$ with those of $(\boldsymbol{x}_i)_{i<i_0}$. This ensures that the residual distribution of $\texttt{Initialize}_1$ is within statistical distance $2^{-\Omega(n)}$ from the residual distribution of the following experiment: Sample $\boldsymbol{x}_i \hookleftarrow D_{\mathbb{Z}^m,S}$ for $i < i_0$, $A^+ = (\boldsymbol{u}^t\|A) \hookleftarrow U(\mathbb{Z}_q^{(m+1)\times n})$ conditioned on $\boldsymbol{x}_i^{+t} \cdot A^+ = \boldsymbol{0}$ for all $i < i_0$, $\boldsymbol{x}_i \hookleftarrow D_{\Lambda^\perp_{-\boldsymbol{u}}(A),S}$ for $i \in [i_0, t+1]$, and $\boldsymbol{b}_j^+ \hookleftarrow U(\mathrm{Span}_{i<i_0}(\boldsymbol{x}_i^+)^\perp)$ for $j \le t - i_0 + 2$. The samplings of the last $\boldsymbol{x}_i^+$'s and those of the $\boldsymbol{b}_j^+$'s being independent, their order can be exchanged. We can now apply Theorem 11 a second time, to postpone the samplings of $(\boldsymbol{x}_i)_{i<i_0}$ after those of the $\boldsymbol{b}_j^+$'s. This gives us that the residual distributions of the above experiment and that of $\texttt{Initialize}_2$ are within statistical distance $2^{-\Omega(n)}$. Overall, we have shown that the residual distributions of $(A, \boldsymbol{u}, (\boldsymbol{b}_j)_j, (v_j)_j, (\boldsymbol{x}_i)_i)$ after $\texttt{Initialize}_1$ and $\texttt{Initialize}_2$ are within exponentially small statistical distance. Hence algorithm $\mathcal{A}$ wins Game_2 with non-negligible advantage.

Now, consider Game_3, which differs from Game_2 only in that $\boldsymbol{x}_{i_0}$ is also sampled from $D_{\Lambda^\perp_{-\boldsymbol{u}'}(A'),S}$.

- $\texttt{Initialize}_3$: Sample $A \hookleftarrow U(\mathbb{Z}_q^{m\times n})$, $\boldsymbol{u} \hookleftarrow U(\mathbb{Z}_q^n)$, $\boldsymbol{b}_j \hookleftarrow U(\mathbb{Z}_q^m)$ and $v_j \hookleftarrow U(\mathbb{Z}_q)$ for $j \le t - i_0 + 2$, $\boldsymbol{x}_i \hookleftarrow D_{\Lambda^\perp_{-\boldsymbol{u}}(A),S}$ for $i > i_0$ and $\boldsymbol{x}_i \hookleftarrow D_{\Lambda^\perp_{-\boldsymbol{u}'}(A'),S}$ for $i \le i_0$

As $\boldsymbol{x}_{i_0}$ is not given to $\mathcal{A}$ at step $\texttt{Input}_3$ and as it is not involved in the challenge distributions this modification does not alter the winning probability of $\mathcal{A}$: algorithm $\mathcal{A}$ also wins Game_3 with non-negligible advantage. Now, we again use Theorem 11 twice, but with $(\boldsymbol{x}_i)_{i\le i_0}$: once for swapping the samplings of these $\boldsymbol{x}_i$'s with A^+ and the $\boldsymbol{b}_j^+$'s, and once for swapping the samplings of A^+ and these $\boldsymbol{x}_i$'s. This shows that algorithm $\mathcal{A}$ wins Game_4 with non-negligible advantage, where Game_4 differs from Game_3 only in its first step, as follows.

- $\texttt{Initialize}_4$: Sample $A \hookleftarrow U(\mathbb{Z}_q^{m\times n})$, $\boldsymbol{u} \hookleftarrow U(\mathbb{Z}_q^n)$, $\boldsymbol{x}_i \hookleftarrow D_{\Lambda^\perp_{-\boldsymbol{u}}(A),S}$ for $i \le t$, and $\boldsymbol{b}_j^+ \hookleftarrow U(\mathrm{Span}_{i\le i_0}(\boldsymbol{x}_i^+)^\perp)$ for $j \le t - i_0 + 2$.

The situation we are in now is very similar to that in the first situation, where $\mathcal{A}$ was supposed to win Game_1. The arguments used in the first situation readily carry over here (up to replacing $\mathrm{Span}_{i<i_0}\boldsymbol{x}_i^+$ and $\mathrm{Span}_{i\ge i_0}\boldsymbol{x}_i^+$ by $\mathrm{Span}_{i\le i_0}\boldsymbol{x}_i^+$ and $\mathrm{Span}_{i>i_0}\boldsymbol{x}_i^+$, respectively). □

5 Projective sampling and public traceability

We now modify the scheme of Section 4 so that the tracing signals can be publicly sampled. For this purpose, we introduce the concept of projective sampling family.

5.1 Projective sampling

Inspired from the notion of projective hash family [23], we propose the notion of projective sampling family in which each sampling function is keyed and, with a projected key, one can simulate the sampling function in a computationally indistinguishable way. Let X be a finite non-empty set. Let $F = (\mathtt{F}_k)_{k \in K}$ be a collection of sampling functions indexed by K, so that $\mathtt{F}_k$ is a sampling function over X, for every $k \in K$. We call $\mathtt{Sam} = (F, K, X)$ a sampling family. We now introduce the concept of projective sampling.

Definition 26 (Projective Sampling). Let $\mathtt{Sam} = (F, K, X)$ be a sampling family. Let J be a finite, non-empty set, and let $\pi : K \to J$ be a (probabilistic) function. Let also $\mathtt{P} = (\mathtt{P}_j)_{j \in J}$ be a collection of sampling functions over X, and D be a distribution over K. Then $\mathtt{PSam} = (F, K, X, \mathtt{P}, J, \pi, D)$ is called a projective sampling family if, with overwhelming probability over the choice of $k, k' \hookleftarrow D$, and given the secret key k and its projected key $\pi(k)$, 1) the distributions obtained using $\mathtt{F}_k$ and $\mathtt{P}_{\pi(k)}$ are computationally indistinguishable, and 2) the distributions obtained using $\mathtt{F}_k$ and $\mathtt{P}_{\pi(k')}$ can be efficiently distinguished.

The first condition means that for $k \hookleftarrow D$, the value $\pi(k)$ "encodes" the sampling distribution of $\mathtt{F}_k$, so that when $\pi(k)$ is made public, the sampled signal $\mathtt{F}_k$ can be publicly simulated by $\mathtt{P}_{\pi(k)}$. The security requirement is very strong because the adversary is not only given the projected key, as in projective hashing, but also the secret key k. We require that sampling signals from the secret key and from its projected key be indistinguishable for the insiders who know the secret key. This is relevant for traitor tracing, as the traitors are system insiders and they possess secret data. The second condition (that we actually do not directly use in our cryptographic application) allows to prevent the trivial solution consisting in setting $\mathtt{P}_{\pi(k)}$ as an efficient sampling function that is independent of k: the simulation signal $\mathtt{P}_{\pi(k)}$ must be specific to k.[3]

5.2 Projective sampling from k-LWE

We construct a set of projective sampling families $(\mathtt{PSam}_i)_{0 \leq i \leq t}$. The parameters are almost identical to the parameters in the `Setup` of the multi-user scheme of Section 4. A further difference, required for simulation purposes in the security proof, is that $\sigma' > \sigma$ must be set $\widetilde{\Omega}(\sqrt{mn} + \pi q)$.

We let $A \hookleftarrow U(\mathbb{Z}_q^{m \times n})$ and $\boldsymbol{u} \hookleftarrow U(\mathbb{Z}_q^n)$ be public parameters. For each i, we define $K_i = (\mathbb{Z}_q^m)^i$ and D_i as the distribution on K_i that samples $k = (\boldsymbol{x}_j)_{j \leq i}$ with $\boldsymbol{x}_j \hookleftarrow D_{\Lambda^{\perp}_{-\boldsymbol{u}}(A), \sigma}$ for all $j \leq i$. The sampling function $\mathtt{F}_{i,k}$ is defined as $U(\mathrm{Span}_{j \leq i}(\boldsymbol{x}_j^+)^{\perp}) + \lfloor \nu_{\alpha q} \rceil^{m+1}$. The projected key $\pi_i(k)$ is defined as follows:

- Sample $H \in \mathbb{Z}_q^{m \times (m-n)}$ uniformly, conditioned on $\mathrm{Im}(A) \subseteq \mathrm{Im}(H)$.
- For each $j \leq i$, define $\boldsymbol{h}_j^t = -\boldsymbol{x}_j^t \cdot H$.
- Finally, set $J = \mathbb{Z}_q^{m \times (m-n)} \times (\mathbb{Z}_q^{m-n})^i$ and set $\pi_i(k) = (H, (\boldsymbol{h}_j)_{j \leq i})$.

We now define the sampling $\mathtt{P}_{i, \pi_i(k)}$ with projected key $\pi_i(k) = (H, (\boldsymbol{h}_j)_{j \leq i})$, as follows:

- Set $H_j = (\boldsymbol{h}_j^t \| H) \in \mathbb{Z}_q^{(m+1) \times (m-n)}$. We have $\boldsymbol{x}_j^{+t} \cdot H_j = \mathbf{0}$ and $\mathrm{Im}(A^+) \subseteq \mathrm{Im}(H_j)$.
- Set $\mathtt{P}_{i, \pi_i(k)} = U\left(\cap_{j \leq i} \mathrm{Im}(H_j)\right) + \lfloor \nu_{\alpha q} \rceil^{m+1}$, with $\cap_{j \leq 0} \mathrm{Im}(H_j) = \mathbb{Z}_q^{m+1}$ by convention. Note that $\cap_{j \leq i} \mathrm{Im}(H_j) \subseteq \mathrm{Span}_{j \leq i}(\boldsymbol{x}_j^+)^{\perp}$.

Theorem 27. *For each $i = 0, \ldots, t$, $\mathtt{PSam}_i$ is a projective sampling family. Concretely, under the (i, S)-LWE$_{\alpha, m}$ hardness assumptions, given the uniformly sampled public*

[3] Another trivial situation occurs when $\pi(k) = k$: the projected key leaks the full information about the original key and one cannot safely publish the projected key.

parameters $(A, \boldsymbol{u})$, the secret key $k = (\boldsymbol{x}_j)_{j\le i} \hookleftarrow D_i$ and its projected key $\pi_i(k) = (H, (\boldsymbol{h}_j)_{j\le i})$, the distributions $\mathtt{F}_{i,k}$ and $\mathtt{P}_{i,\pi_i(k)}$ are indistinguishable. Moreover, they are both indistinguishable from $U(\mathrm{Im}(A^+)) + \lfloor \nu_{\alpha q} \rceil^{m+1}$. Finally, with overwhelming probability, the distributions $\mathtt{F}_{i,k}$ and $\mathtt{P}_{i,\pi_i(k')}$ can be efficiently distinguished, when k' is independently sampled from D_i.

Proof. For the last statement, observe that with overwhelming probability, the secret key k' contains an $\boldsymbol{x}'_j \in \mathbb{Z}_q^m$ that does not belong to $\mathrm{Span}_{j\le i}(\boldsymbol{x}_j)$ (by Lemma 12). In that case, taking the inner product of all $\boldsymbol{x}'_j$'s of k' with a sample from $\mathtt{P}_{i,\pi_i(k')}$ gives small residues modulo q, whereas one of the inner products of the $\boldsymbol{x}'_j$'s with a sample from $\mathtt{F}_{i,k}$ will be uniform modulo q.

We now consider the first statement. From the hardness of (i, S)-$\mathrm{LWE}_{\alpha,m}$, given k, the distributions

$$\mathtt{F}_{i,k} = U(\mathrm{Span}_{j\le i}(\boldsymbol{x}_j^+)^\perp) + \lfloor \nu_{\alpha q} \rceil^{m+1} \quad \text{and} \quad U(\mathrm{Im}(A^+)) + \lfloor \nu_{\alpha q} \rceil^{m+1}$$

are indistinguishable. Further, given $k = (\boldsymbol{x}_j)_{j\le i}$, the projected key $\pi_i(k) = (H, (\boldsymbol{h}_j)_{j\le i})$ can be sampled from D_i. Therefore, given both k and $\pi_i(k)$, the distributions $\mathtt{F}_{i,k}$ and $U(\mathrm{Im}(A^+)) + \lfloor \nu_{\alpha q} \rceil^{m+1}$ remain indistinguishable.

Now, we have $\mathrm{Im}(A^+) \subseteq \cap_{j\le i}\mathrm{Im}(H_j) \subseteq (\mathrm{Span}_{j\le i}(\boldsymbol{x}_j^+))^\perp$. Hence:

$$\begin{aligned} U(\mathrm{Im}(A^+)) + U(\cap_{j\le i}\mathrm{Im}(H_j)) &= U(\cap_{j\le i}\mathrm{Im}(H_j)), \\ U(\mathrm{Span}_{j\le i}(\boldsymbol{x}_j^+)^\perp) + U(\cap_{j\le i}\mathrm{Im}(H_j)) &= U(\mathrm{Span}_{j\le i}(\boldsymbol{x}_j^+)^\perp). \end{aligned}$$

We note that given $\boldsymbol{h}_1, \ldots, \boldsymbol{h}_i$, one can efficiently sample from $U(\cap_{j\le i}\mathrm{Im}(H_j))$. Therefore, under the hardness of (i, S)-$\mathrm{LWE}_{\alpha,m}$, this shows that $\mathtt{F}_{i,k}$, $\mathtt{P}_{i,\pi_i(k)}$ and $U(\mathrm{Im}(A^+))+ \lfloor \nu_{\alpha q} \rceil^{m+1}$ are indistinguishable. ☐

5.3 Public traceability from projective sampling

In the scheme of Section 4, the tracing key $tk = (T, (\mathcal{U}_i, \boldsymbol{x}_i)_{i\le N})$ must be kept secret, as it would reveal the secret keys of the users. The tracing signals are samples from $U(\mathrm{Span}_{j\le i}(\boldsymbol{x}_j^+)^\perp) + \lfloor \nu_{\alpha q} \rceil^{m+1}$, which exactly matches $\mathtt{F}_{i,k}$. By publishing the projected key $\pi_i(k)$, anyone can use the projective sampling $\mathtt{P}_{i,\pi_i(k)}$: by Theorem 27, given $(k, \pi_i(k))$, $\mathtt{F}_{i,k}$ and $\mathtt{P}_{i,\pi_i(k)}$ are indistinguishable and they are both indistinguishable from the original sampling $U(\mathrm{Im}(A^+)) + \lfloor \nu_{\alpha q} \rceil^{m+1}$. We are thus almost done with public traceability.

However, a subtle point is that we have to use all the projective samplings $(\mathtt{P}_{i,\pi_i(k)})$ for transforming the secret tracing to the public tracing: all the projected keys $(\boldsymbol{h}_j)_{j\le N}$ should be published. Because the keys k in $\mathtt{F}_{i,k}$ are not independent, it could occur that the adversary exploits a projected key $\pi_i(k)$ for distinguishing $\mathtt{P}_{i',\pi_{i'}(k')}$ from the original signals. To handle this, we prove that, given $(\boldsymbol{x}_j)_{j\le i}$ and all the keys $(\boldsymbol{h}_j)_{j\le N}$, the adversary cannot distinguish $\mathtt{P}_{i,\pi_i(k)}$ from the original signals. For this purpose, we exploit a technique from [28] to simulate $(\boldsymbol{h}_j)_{i<j\le N}$ from the public information.

Theorem 28. *Set $i \le t$. Under the (i, S)-$\mathrm{LWE}_{\alpha,m}$ and the $\mathrm{LWE}'_{\alpha,m}$ hardness assumptions, given the secret key $k = (\boldsymbol{x}_j)_{j\le i}$ and the projected keys $(H, (\boldsymbol{h}_j)_{j\le N})$, the following two distributions are indistinguishable*

$$\mathtt{P}_{i,\alpha(k)} = U(\cap_{j\le i}\mathrm{Im}(H_j)) + \lfloor \nu_{\alpha q} \rceil^{m+1} \quad \textit{and} \quad U(\mathrm{Im}(A^+)) + \lfloor \nu_{\alpha q} \rceil^{m+1}.$$

Proof. Assume a ppt attacker is given $(\boldsymbol{x}_j)_{j\leq i}$ (with the $\boldsymbol{x}_j$'s independently sampled from $D_{\Lambda^\perp_{-\boldsymbol{u}}(A),\sigma}$) and all the projected keys $(\boldsymbol{h}_j)_{j\leq N}$. We are to prove that, under the (i,S)-$\mathrm{LWE}_{\alpha,m}$ and $\mathrm{LWE}'_{\alpha,m}$ hardness assumptions, it cannot distinguish between the distributions (over $\mathbb{Z}_q^{m+1}$)

$$U(\mathrm{Im}(A^+)) + \lfloor \nu_{\alpha q} \rceil^{m+1} \quad \text{and} \quad \mathsf{P}_{i,\pi_i(k)} = U(\cap_{j\leq i}\mathrm{Im}(H_j)) + \lfloor \nu_{\alpha q} \rceil^{m+1}.$$

We proceed by a sequence of games.
Game_0: This is the above distinguishing game. We let ε_0 denote the adversary's distinguishing advantage. The goal is to show that ε_0 is negligible.
Game_1: In this second game, we sample $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_i$ from $D_{\Lambda^\perp_{-\boldsymbol{u}}(A),\sigma}$ as in **Game**$_0$, but the $\boldsymbol{x}_j$'s for $j>i$ are sampled uniformly in $\mathbb{Z}_q^n$, conditioned on $\boldsymbol{x}_j^t \cdot A = -\boldsymbol{u}^t$. The $\boldsymbol{h}_j$'s for $j>i$ are modified accordingly, but the rest is as in Game_0. We let ε_1 denote the adversary's distinguishing advantage.

The main point is that in Game_1, no secret information is required for sampling the projected keys $\boldsymbol{h}_j$'s for $j>i$.

Lemma 29. *Under the* $\mathrm{LWE}'_{\alpha,m}$ *hardness assumption, the quantity* $|\varepsilon_1 - \varepsilon_0|$ *is negligible.*

Proof. Our aim is to reduce $\mathrm{LWE}'_{\alpha,m+1}$ to distinguishing Game_1 and Game_0. Assume we have the following multiple LWE' input $(B,\boldsymbol{y}_{i+1},\ldots,\boldsymbol{y}_N)$ where $B \hookleftarrow U(\mathbb{Z}_q^{m\times n})$, and $\boldsymbol{y}_j = B\boldsymbol{s}_j + \boldsymbol{e}_j$ with $\boldsymbol{s}_j \hookleftarrow U(\mathbb{Z}_q^n)$ and either $\boldsymbol{e}_j \hookleftarrow U(\mathbb{Z}_q^m)$ for all j, or $\boldsymbol{e}_j \hookleftarrow D_{\mathbb{Z}^m,\alpha q}$ for all j. Our goal is to exploit a distinguisher between Game_0 and Game_1 to decide whether the $\boldsymbol{e}_j$'s are Gaussian or uniform. We simulate Game_1 and Game_0 as follows (depending on the nature of $\boldsymbol{e}_i$):

- Sample $A \in \mathbb{Z}_q^{m\times n}$ and $T \in \mathbb{Z}^{m\times m}$ such that A is uniform conditioned on $B^t \cdot A = 0$ and T is a full-rank basis of $\Lambda^\perp(A)$ satisfying $\|T\| \leq O(\sqrt{mn\log q \log m})$. This can be performed in ppt using [28, Le. 4].
- Define H as a randomized basis of the kernel of B. It is $m\times(m-n)$ with probability $2^{-\Omega(n)}$. The distribution of the pair (A,H) is within statistical distance $2^{-\Omega(n)}$ of its distribution in Game_0 and Game_1.
- Sample $\boldsymbol{u} \hookleftarrow U(\mathbb{Z}_q^n)$ and sample the keys $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_i \hookleftarrow D_{\Lambda^\perp_{-\boldsymbol{u}}(A),S}$ by using the trapdoor matrix T (this is why σ' must be set sufficiently large). Compute $\boldsymbol{h}_j^t = -\boldsymbol{x}_j^t \cdot H$ for $j\leq i$.
- Using linear algebra, find $\boldsymbol{c}$ such that $\boldsymbol{c}^t \cdot A = \boldsymbol{u}^t$. For each $j\in[i+1,N]$:
  - Compute $\boldsymbol{u}_j^t = \boldsymbol{y}_j^t \cdot A$. Since $\boldsymbol{y}_j = B\cdot \boldsymbol{s}_j + \boldsymbol{e}_j$, we have $\boldsymbol{u}_j^t = \boldsymbol{e}_j^t \cdot A$ (although we would prefer $\boldsymbol{u}^t = \boldsymbol{e}_j^t \cdot A$).
  - Sample $\boldsymbol{e}'_j \hookleftarrow \boldsymbol{c} - \boldsymbol{y}_j + D_{\Lambda^\perp(A),S_2,-\boldsymbol{c}+\boldsymbol{y}_j}$ where $S_2 = \sqrt{SS^t - \alpha^2q^2I_m}$ (these are diagonal matrices), using T. Since $\boldsymbol{y}_j - \boldsymbol{e}_j \in \Lambda^\perp(A)$, we can rewrite the latter as $\boldsymbol{e}'_j \hookleftarrow \boldsymbol{c} - \boldsymbol{e}_j + D_{\Lambda^\perp(A),S_2,-\boldsymbol{c}+\boldsymbol{e}_j}$.
  - Compute $\boldsymbol{z}_j = \boldsymbol{y}_j + \boldsymbol{e}'_j$. We now have $(\boldsymbol{e}_j^t + \boldsymbol{e}_j'^t)\cdot A = \boldsymbol{z}_j^t \cdot A = \boldsymbol{c}^t \cdot A = \boldsymbol{u}^t$.
  - Set $\boldsymbol{h}_j^t = -\boldsymbol{z}_j^t \cdot H$. Note that $\boldsymbol{h}_j^t = -(\boldsymbol{e}_j^t + \boldsymbol{e}_j'^t)\cdot H$.
- Return $A, \boldsymbol{u}, H, (\boldsymbol{x}_j)_{j\leq i}$ and $(\boldsymbol{h}_j)_{j\leq N}$.

We observe that for each $j\in[i+1,N]$, we have $\boldsymbol{z}_j = \boldsymbol{y}_j + \boldsymbol{e}'_j = B\cdot\boldsymbol{s}_j + (\boldsymbol{e}_j + \boldsymbol{e}'_j)$. We consider two cases.

- When $\boldsymbol{e}_j \hookleftarrow D_{\mathbb{Z}^m,\alpha q}$, the residual distribution of $D_{\Lambda^\perp(A),S_2,-\boldsymbol{c}+\boldsymbol{e}_j}$ is within negligible statistical distance to $D_{\Lambda^\perp(A),S,-\boldsymbol{c}}$; this is provided by Lemma 4, whose assumptions are satisfied (thanks to the second lower bound on σ') and to Lemma 9; consequently, the residual distribution of $\boldsymbol{e}_j + \boldsymbol{e}'_j$ is negligibly close to the distribution $\boldsymbol{c} + D_{\Lambda^\perp(A),S,-\boldsymbol{c}}$, and hence the distribution of $\boldsymbol{z}_j$ is statistically close to $D_{\Lambda^\perp_{\boldsymbol{u}}(A),S}$. Overall, the data available to the adversary follows the same distributions as in Game_0, up to negligible statistical distance.
- When $\boldsymbol{e}_j \hookleftarrow U(\mathbb{Z}_q^m)$, the residual distribution of $\boldsymbol{z}_j$ is uniform (by adapting the argument above). The data available follows the same distributions as in Game_1, up to negligible statistical distance.

This completes the proof of the lemma. □

We note that, in Game_1, the $\boldsymbol{h}_j$'s can be sampled publicly from the available data. Therefore, from Theorem 27, under the (i, S)-$\mathrm{LWE}_{\alpha,m}$ hardness assumptions, the advantage ε_1 is negligible. □

Semantic security of the updated scheme. We modify the public information of the scheme of Section 4, so that we can use the set of projective sampling families described above. For this aim, we simply add the projected key $(H, (\boldsymbol{h}_i)_{i \le N})$ to the public key. The scheme becomes publicly traceable because the tracing signals can be sampled from the projected keys, as explained above. Finally, as the public key has been modified, we should prove that the knowledge of these projected keys provides no significant advantage for an adversary towards breaking the semantic security of the encryption scheme. Fortunately, the semantic security directly follows from Theorem 28, for the particular case of $i = 0$.

Acknowledgements. We thank M. Abdalla, D. Augot, R. Bhattacharrya, L. Ducas, V. Guleria, G. Hanrot, F. Laguillaumie, K. T. T. Nguyen, G. Quintin, O. Regev, H. Wang for helpful discussions. The authors were partly supported by the LaBaCry MERLION grant, the Australian Research Council Discovery Grant DP110100628, the ANR-09-VERSO-016 BEST and ANR-12-JS02-0004 ROMAnTIC Projects, the INRIA invited researcher scheme, the Singapore National Research Foundation Research Grant NRF-CRP2-2007-03, the Singapore MOE Tier 2 research grant MOE2013-T2-1-041, the LIA Formath Vietnam and the ERC Starting Grant ERC-2013-StG-335086-LATTAC.

References

1. M. Abdalla, A. W. Dent, J. Malone-Lee, G. Neven, D. H. Phan, and N. P. Smart. Identity-based traitor tracing. In *Proceedings of PKC*, volume 4450 of *LNCS*, pages 361–376. Springer, 2007.
2. D. Aggarwal and O. Regev. A note on discrete gaussian combinations of lattice vectors, 2013. Draft. Available at `http://arxiv.org/pdf/1308.2405v1.pdf`.
3. S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *Proc. of EUROCRYPT*, volume 6110 of *LNCS*, pages 553–572. Springer, 2010. Full version available from the authors upon request.
4. S. Agrawal, D. Boneh, and X. Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In *Proc. of CRYPTO*, volume 6223 of *LNCS*, pages 98–115. Springer, 2010.
5. S. Agrawal, C. Gentry, S. Halevi, and A. Sahai. Sampling discrete gaussians efficiently and obliviously. In *Proc. of ASIACRYPT (1)*, volume 8269 of *LNCS*, pages 97–116. Springer, 2013.
6. M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proc. of STOC*, pages 99–108. ACM, 1996.

7. M. Ajtai. Generating hard instances of the short basis problem. In *Proc. of ICALP*, volume 1644 of *LNCS*, pages 1–9. Springer, 1999.
8. J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. *Theor. Comput. Science*, 48(3):535–553, 2011.
9. S. Bai, A. Langlois, T. Lepoint, D. Stehlé, and R. Steinfeld. Improved security proofs in lattice-based cryptography: using the Rényi divergence rather than the statistical distance. *IACR Cryptology ePrint Archive*, 2015:483, 2015.
10. O. Billet and D. H. Phan. Efficient Traitor Tracing from Collusion Secure Codes. In *Proc. of ICITS*, volume 5155 of *LNCS*, pages 171–182. Springer, 2008.
11. D. Boneh and M. K. Franklin. An efficient public key traitor tracing scheme. In *Proc. of CRYPTO*, volume 1666 of *LNCS*, pages 338–353. Springer, 1999.
12. D. Boneh and D. M. Freeman. Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures. In *Proc. of PKC*, volume 6571 of *LNCS*, pages 1–16. Springer, 2011. Full version available at `http://eprint.iacr.org/2010/453.pdf`.
13. D. Boneh and M. Naor. Traitor tracing with constant size ciphertext. In *Proc. of ACM CCS*, pages 501–510. ACM, 2008.
14. D. Boneh, A. Sahai, and B. Waters. Fully collusion resistant traitor tracing with short ciphertexts and private keys. In *Proc. of EUROCRYPT*, volume 4004 of *LNCS*, pages 573–592. Springer, 2006.
15. D. Boneh and B. Waters. A fully collusion resistant broadcast, trace, and revoke system. In *Proc. of ACM CCS*, pages 211–220. ACM, 2006.
16. D. Boneh and M. Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In *Proc. of CRYPTO*, volume 8616 of *LNCS*, pages 480–499. Springer, 2014.
17. Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. Cryptology ePrint Archive, Report 2013/642, 2013. `http://eprint.iacr.org/`.
18. Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. Classical hardness of learning with errors. In *STOC*, pages 575–584. ACM, 2013.
19. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *Proc. of EUROCRYPT*, volume 6110 of *LNCS*, pages 523–552. Springer, 2010.
20. H. Chabanne, D. H. Phan, and D. Pointcheval. Public traceability in traitor tracing schemes. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 542–558. Springer, May 2005.
21. B. Chor, A. Fiat, and M. Naor. Tracing traitors. In *Proc. of CRYPTO*, volume 839 of *LNCS*, pages 257–270. Springer, 1994.
22. B. Chor, A. Fiat, M. Naor, and B. Pinkas. Tracing traitors. *IEEE Trans. Inf. Th.*, 46(3):893–910, 2000.
23. R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 45–64. Springer, April / May 2002.
24. N. Fazio, A. Nicolosi, and D. H. Phan. Traitor tracing with optimal transmission rate. In *Proc. of ISC*, volume 4779 of *LNCS*, pages 71–88. Springer, 2007.
25. A. Fiat and M. Naor. Broadcast encryption. In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 480–491. Springer, August 1994.
26. S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *Proc. of FOCS*, pages 40–49. IEEE Computer Society Press, 2013.
27. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proc. of STOC*, pages 197–206. ACM, 2008. Full version available at `http://eprint.iacr.org/2007/432.pdf`.
28. S. D. Gordon, J. Katz, and V. Vaikuntanathan. A group signature scheme from lattice assumptions. In *Proc. of ASIACRYPT*, volume 2647 of *LNCS*, pages 395–412. Springer, 2010.
29. A. Kiayias and S. Pehlivanglu. *Encryption For Digital Content*. Springer, 2010.
30. A. Kiayias and M. Yung. On crafty pirates and foxy tracers. In *Proc. of DRM Workshop*, volume 2320 of *LNCS*, pages 22–39. Springer, 2001.
31. A. Kiayias and M. Yung. Self protecting pirates and black-box traitor tracing. In *Proc. of CRYPTO*, volume 2139 of *LNCS*, pages 63–79. Springer, 2001.
32. A. Kiayias and M. Yung. Breaking and repairing asymmetric public-key traitor tracing. In *Digital Rights Management Workshop*, pages 32–50, 2002.
33. A. Kiayias and M. Yung. Traitor tracing with constant transmission rate. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 450–465. Springer, April / May 2002.

34. P. N. Klein. Finding the closest lattice vector when it's unusually close. In *Proc. of SODA*, pages 937–941. ACM, 2000.
35. H. Komaki, Y. Watanabe, G. Hanaoka, and H. Imai. Efficient asymmetric self-enforcement scheme with public traceability. In Kwangjo Kim, editor, *PKC 2001*, volume 1992 of *LNCS*, pages 225–239. Springer, February 2001.
36. K. Kurosawa and Y. Desmedt. Optimum traitor tracing and asymmetric schemes. In *Proc. of EUROCRYPT*, LNCS, pages 145–157. Springer, 1998.
37. K. Kurosawa and T. Yoshida. Linear code implies public-key traitor tracing. In David Naccache and Pascal Paillier, editors, *PKC 2002*, volume 2274 of *LNCS*, pages 172–187. Springer, February 2002.
38. A. Langlois, D. Stehlé, and R. Steinfeld. GGHLite: More efficient multilinear maps from ideal lattices. In *Proc. of EUROCRYPT*, LNCS, pages 239–256. Springer, 2014.
39. B. Libert and D. Stehlé. Fully secure functional encryption for inner products, from standard assumptions, 2015. Available at `http://eprint.iacr.org/2015/608`.
40. S. Ling, D. H. Phan, D. Stehlé, and R. Steinfeld. Hardness of k-LWE and applications in traitor tracing. In *Proc. of CRYPTO*, volume 8616 of *LNCS*, pages 315–334. Springer, 2014.
41. V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. *J. ACM*, 60(6):43, 2013.
42. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Proc. of EUROCRYPT*, volume 7237 of *LNCS*, pages 700–718. Springer, 2012.
43. D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput*, 37(1):267–302, 2007.
44. D. Micciancio and O. Regev. Lattice-based cryptography. In *Post-Quantum Cryptography, D. J. Bernstein, J. Buchmann, E. Dahmen (Eds)*, pages 147–191. Springer, 2009.
45. D. Naor, M. Naor, and J. Lotspiech. Revocation and tracing schemes for stateless receivers. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 41–62. Springer, August 2001.
46. M. Naor and B. Pinkas. Efficient trace and revoke schemes. In Yair Frankel, editor, *FC 2000*, volume 1962 of *LNCS*, pages 1–20. Springer, February 2000.
47. M. Naor and B. Pinkas. Efficient trace and revoke schemes. In *Proc. of Financial Cryptography*, volume 1962 of *LNCS*, pages 1–20. Springer, 2000.
48. A. O'Neill, C. Peikert, and B. Waters. Bi-deniable public-key encryption. In *Proc. of CRYPTO*, volume 6841 of *LNCS*, pages 525–542. Springer, 2011.
49. C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *Proc. of STOC*, pages 333–342. ACM, 2009.
50. C. Peikert. An efficient and parallel Gaussian sampler for lattices. In *Proc. of CRYPTO*, volume 6223 of *LNCS*, pages 80–97. Springer, 2010.
51. C. Peikert, A. Shelat, and A. Smith. Lower bounds for collusion-secure fingerprinting. In *Proc. of SODA*, pages 472–479, 2003.
52. C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In *Proc. of STOC*, pages 187–196. ACM, 2008.
53. B. Pfitzmann. Trials of traced traitors. In *Information Hiding*, volume 1174 of *LNCS*, pages 49–64. Springer, 1996.
54. B. Pfitzmann and M. Waidner. Asymmetric fingerprinting for larger collusions. In *ACM CCS 97*, pages 151–160. ACM Press, April 1997.
55. D. H. Phan, R. Safavi-Naini, and D. Tonien. Generic construction of hybrid public key traitor tracing with full-public-traceability. In *Proc. of ICALP (2)*, volume 4052 of *LNCS*, pages 264–275. Springer, 2006.
56. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proc. of STOC*, pages 84–93. ACM, 2005.
57. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009.
58. O. Regev. The learning with errors problem, 2010. Invited survey in CCC 2010, available at `http://www.cims.nyu.edu/~regev/`.
59. A. Silverberg, J. Staddon, and J. L. Walker. Efficient traitor tracing algorithms using list decoding. In *Proc. of ASIACRYPT*, volume 2248 of *LNCS*, pages 175–192. Springer, 2001.
60. T. Sirvent. Traitor tracing scheme with constant ciphertext rate against powerful pirates. In Daniel Augot, Nicolas Sendrier, and Jean-Pierre Tillich, editors, *Workshop on Coding and Cryptography—WCC '07*, pages 379–388, April 2007.
61. D. R. Stinson and R. Wei. Combinatorial properties and constructions of traceability schemes and frameproof codes. *SIAM J. Discrete Math.*, 11(1):41–53, 1998.
62. D. R. Stinson and R. Wei. Key preassigned traceability schemes for broadcast encryption. In *Proc. of SAC*, volume 1556 of *LNCS*, pages 144–156. Springer, 1998.

63. G. Tardos. Optimal probabilistic fingerprint codes. *J. ACM*, 55(2), 2008.
64. Y. Watanabe, G. Hanaoka, and H. Imai. Efficient asymmetric public-key traitor tracing without trusted agents. In David Naccache, editor, *CT-RSA 2001*, volume 2020 of *LNCS*, pages 392–407. Springer, April 2001.