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
lingsan@ntu.edu.sg – http://www.ntu.edu.sg/home/lingsan/

[2] CNRS, Laboratoire LIP (U. Lyon, CNRS, ENS Lyon, INRIA, UCBL),
46 Allée d'Italie, 69364 Lyon Cedex 07, France.
damien.stehle@gmail.com – http://perso.ens-lyon.fr/damien.stehle

[3] CNRS, Laboratoire LAGA (U. Paris 8, U. Paris 13),
2 Rue de la Liberté, 93200 Saint-Denis, France.
duong-hieu.phan@ens.fr – http://www.di.ens.fr/users/phan/

[4] Monash University,
Clayton VIC 3800, Australia, France.
ron.steinfeld@monash.edu – http://users.monash.edu.au/~rste/

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 k-LWE problem and the projective sampling family are quite general that they may have applications in other areas.

Keywords. Lattice-based cryptography, Traitor tracing, LWE, Public traceability, Provable security.

1 Introduction

Since the pioneering work of Ajtai [4], 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 [33, 44]). 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 [42, 43], 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. observed in [21] 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. [16], much work has been devoted to devising efficient and secure traitor tracing schemes. The Boneh-Franklin scheme [8] is one of the first algebraic construction but it can still be considered as the referenced algebraic transformation from the standard ElGamal public key encryption into traitor tracing. This transformation induces a linear loss in efficiency, in the bound of 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 [8, 34, 27, 46, 7, 12]. We refer to [23] for a detailed introduction to this rich topic. Also, in Appendix A.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 [43, 36, 13].

The scheme is the extension, described above, of the dual-Regev LWE-based encryption scheme from [21] 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 [9], 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 [5] (see also Lemma 2). We adapt the Micciancio-Peikert framework from [31] to sampling a Gaussian $X \in \mathbb{Z}^{m \times n}$ along with a short basis for the lattice $\ker(X) = \{\boldsymbol{b} \in \mathbb{Z}^m : \boldsymbol{b}^t X = \boldsymbol{0}\}$. Kernel lattices also play an important role in the re-randomization analysis of the recent lattice-based multilinear map scheme of Garg et al. [20], and we believe that our new trapdoor generation tool for such lattices is likely find additional applications in future. We also remark that 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 [9]: from an exponential loss in k to a polynomial loss in k. Consequently, their linearly homomorphic signatures and ordinary signature schemes enjoy much better efficiency/security trade-offs.

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 [8]. 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 as efficient as standard LWE-based encryption. 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 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.

To obtain public traceability and inspired from the notion of projective hash family [18], 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.

Independently, our new lattice tools may have applications in other areas. The k-LWE problem has a similar flavour to the Extended-LWE problem from [35]. It would be interesting to exhibit reductions between these problems. On a closely-related topic, it seems our sampling of a random Gaussian integer matrix X together with a short basis of $\ker(X)$ is compatible with the hardness proof of Extended-LWE from [13]. In particular, it should be possible to use it as an alternative to [13, Def 4.5] in the proof of [13, Le 4.7], to show that Extended-LWE remains hard with many hints independently sampled from discrete Gaussians.

2 Preliminairies

We first recall some basic notations. 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 let $\mathrm{Im}(A)$ denote the set $\{A\boldsymbol{s} : \boldsymbol{s} \in \mathbb{Z}_q^n\} \subseteq \mathbb{Z}_q^m$. For $X \subseteq \mathbb{Z}_q^m$, we let $\mathrm{Span}(X)$ denote the set of all linear combinations of 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 $\varDelta(D_1, D_2)$. 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. We define the function $\rho_{S,\boldsymbol{c}}(\boldsymbol{b}) = \exp(-\pi\|S^{-1}(\boldsymbol{b} - \boldsymbol{c})\|^2)$ for any invertible $S \in \mathbb{R}^{m \times m}$ and $\boldsymbol{c} \in \mathbb{R}^m$. For $S = sI_m$, we write $\rho_{s,\boldsymbol{c}}$, 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 Euclidean lattices and discrete Gaussian distributions

A lattice is a set of the form $\{\sum_{i \leq n} x_i \boldsymbol{b}_i : x_i \in \mathbb{Z}\}$ 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 n-th 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 $\boldsymbol{0}$ contains n linearly independent vectors of L. The smoothing parameter of L is defined as $\eta_\varepsilon(L) = \min\{r > 0 : \rho_{1/r}(\widehat{L} \setminus \boldsymbol{0}) \leq \varepsilon\}$ 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 [32, Le. 3.3] that $\eta_\varepsilon(L) \leq \sqrt{\ln(2n(1+1/\varepsilon))/\pi} \cdot \lambda_n(L)$ for all $\varepsilon \in (0,1)$ and n-dimensional lattice 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. [21] gave an algorithm to sample from $D_{L,S,\boldsymbol{c}}$, which is precised in Lemma 20 in Appendix B. The basic results on lattice are also given in Appendix B.

We extensively use q-ary lattices. The q-ary lattice associated to $A \in \mathbb{Z}_q^{m \times n}$ is defined as $\varLambda^\perp(A) = \{\boldsymbol{x} \in \mathbb{Z}^m : \boldsymbol{x}^t \cdot A = \boldsymbol{0} \bmod q\}$. 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 $\varLambda_{\boldsymbol{u}}^\perp(A)$ as the coset $\{\boldsymbol{x} \in \mathbb{Z}^m : \boldsymbol{x}^t \cdot A = \boldsymbol{u}^t \bmod q\}$ of $\varLambda^\perp(A)$.

2.2 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 $\varLambda^\perp(A)$. The following lemma provides a probabilistic bound on the smoothing parameter of $\varLambda^\perp(A)$.

Lemma 1 (Adapted from [21, Le. 5.3]). *Let q be prime and m, n integers with $m \geq 2n$ and $\varepsilon > 0$, then $\eta_\varepsilon(\varLambda^\perp(A)) \leq 4q^{\frac{n}{m}}\sqrt{\log(2m(1+1/\varepsilon))/\pi}$, for all except a fraction $2^{-\varOmega(n)}$ 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 $\varLambda^\perp(A)$ (see [5, 6, 37, 31]).

Lemma 2 (Adapted from [6, 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 $2^{-\varOmega(n)}$ from $U(\mathbb{Z}_q^{m \times n})$; the rows of T form a basis of $\varLambda^\perp(A)$; 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 and $\boldsymbol{u} \in \mathbb{Z}_q^n$, we define the distribution $D_{\Lambda_{\boldsymbol{u}}^\perp(A),S}$ as $\boldsymbol{c} + D_{\Lambda^\perp(A),S,-\boldsymbol{c}}$, where $\boldsymbol{c}$ is any vector of $\mathbb{Z}^m$ such that $\boldsymbol{c}^t \cdot A = \boldsymbol{u}^t \bmod q$. A sample $\boldsymbol{x}$ from $D_{\Lambda_{\boldsymbol{u}}^\perp(A),S}$ can be obtained using Lemma 20 along with the short basis of $\Lambda^\perp(A)$ provided by Lemma 2. Boneh and Freeman [9] showed how to efficiently obtain the residual distribution of $(A, \boldsymbol{x})$ without relying on Lemma 2.

Theorem 3 (Adapted from [9, 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 $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k \in \mathbb{Z}_q^n$ 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 $2^{-\Omega(n)}$.*

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

This statement generalizes [9, Th. 4.3] in three ways. First, the latter corresponds to the special case corresponding to taking all the $\boldsymbol{u}_i$'s equal to $\mathbf{0}$. Generalizing to arbitrary $\boldsymbol{u}_i$'s does not add any extra complication in the proof of [9, Th. 4.3], but is important for our constructions. Second, the condition on m is less restrictive (the corresponding assumption in [9, Th. 4.3] is that $m \geq \max(2n\log q, 2k)$). To allow for such small values of m, one has to refine the bound on the smoothing parameter of the $\Lambda^\perp(A)$ lattice (namely, we use Lemma 1). Third, we allow for a non-spherical Gaussian distribution, which seems needed in our generalized Micciancio-Peikert trapdoor gadget used in the reduction from LWE to k-LWE in Section 3.2.

We also use the following result on the probability of the Gaussian vectors $\boldsymbol{x}_i$ from Theorem 3 being linearly independent over $\mathbb{Z}_q$.

Lemma 4 (Adapted from [9, Le. 4.5]). *With the notations and assumptions of Theorem 3, 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 $2^{-\Omega(n)}$.*

2.3 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 [42, 43], 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 [43] 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 [36, 13], 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 $\mathbb{T}^m$)

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

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., [32, 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 will be most 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 [37] 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 [9].

Definition 5. Let $k \leq m$ and $S \in \mathbb{R}^{m\times m}$ invertible. The (k, S)-LWE$_{\alpha,m}$ problem is as follows: Given $A \hookleftarrow U(\mathbb{Z}_q^{m\times n}), \boldsymbol{u} \hookleftarrow U(\mathbb{Z}_q^n)$ and $\boldsymbol{x}_i \hookleftarrow D_{\Lambda^\perp_{-\boldsymbol{u}}(A),S}$ for $i \leq k$, the goal is to distinguish between the distributions

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

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 $\approx \alpha\sigma_1(S)$, 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 $D_{\Lambda^\perp((\boldsymbol{u}^t\|A)),S}$ 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 9 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 [9].

In the proof of the hardness of (k, S)-LWE problem, we rely on a gadget integral matrix G that has the following properties: its first rows have Gaussian distributions, it is unimodular and its inverse is small. We build such a gadget matrix by extending Ajtai's simultaneous sampling of a random q-ary lattice with a short basis [5] (see also Lemma 2) to kernel lattices. More precisely, we adapt the Micciancio-Peikert framework [31] to sampling a Gaussian $X \in \mathbb{Z}^{m\times n}$ along with a short basis for the lattice $\ker(X) = \{\boldsymbol{b} \in \mathbb{Z}^m : \boldsymbol{b}^t X = \boldsymbol{0}\}$.

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

The Micciancio-Peikert construction 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 $A^t \cdot D_{\mathbb{Z}^m,\sigma} \bmod q$ 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 (given in appendix) relies on [2], which improves over a similar result from [3]. 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 is mostly cosmetic.

Lemma 6. *Let $m \geq n \geq 100$ 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 $X \hookleftarrow D_{\mathbb{Z},\sigma_1}^{m\times n}$. 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*

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

The proof of this lemma is given in appendix. We now adapt the Micciancio-Peikert trapdoor construction to kernel lattices.

Theorem 7. *Let $n, m_1, \sigma_1, \sigma_2$ be as in Lemma 6, 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), and σ_1, σ_2 as inputs, 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 $D_{\mathbb{Z},\sigma_1}^{m_1 \times n} \times D_{\mathbb{Z},\sigma_2}^{m_2 \times n}$,*
- *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 from $D_{\mathbb{Z},\sigma_1}^{m_1 \times n}$, using Lemma 20. We run m_2 times the algorithm from Lemma 6, 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

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

The result then follows from Lemma 21 (to bound the norms of the rows of X_1) and elementary computations. □

Our gadget matrix G is U^{-t}. In the following corollary, we summarize the properties we will use.

Corollary 8. *Let $n, m_1, m_2, m, \sigma_1, \sigma_2$ be as in Theorem 7. There exists a ppt algorithm that given n, m_1, m_2 (in unary), and σ_1, σ_2 as inputs, returns $G \in \mathbb{Z}^{m \times m}$ such that:*

- *the top $n \times m$ submatrix of G is within statistical distance $2^{-\Omega(n)}$ of $D_{\mathbb{Z},\sigma_1}^{n \times m_1} \times D_{\mathbb{Z},\sigma_2}^{n \times m_2}$,*
- *we have $|\det G| = 1$ and $\|G^{-1}\| \leq O(\sqrt{nm_2}\sigma_2)$, with probability $1 - 2^{-\Omega(n)}$.*

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 9. *There exists $c > 0$ such that the following holds for $k = n/(c \log n)$. Let m, q, σ, σ' be such that $\sigma \geq \Omega(n)$, $\sigma' \geq \Omega(n^3\sigma^2/\log n)$, $q \geq \Omega(\sigma'\sqrt{\log m})$ is prime, and $m \geq \Omega(n \log q)$ (e.g., $\sigma = \Theta(n)$, $\sigma' = \Theta(n^5/\log n)$, $q = \Theta(n^5)$ and $m = \Theta(n \log n)$). Then there exists a probabilistic polynomial-time reduction from $\mathrm{LWE}_{m+1,\alpha}$ in dimension n to (k, S)-$\mathrm{LWE}_{m+2n,\alpha'}$ in dimension $4n$, with $\alpha' = \Omega(mn^{3/2}\sigma\sigma'\alpha)$ and $S = \left[\begin{array}{c|c} \sigma \cdot I_{m+n} & 0 \\ \hline 0 & \sigma' \cdot I_n \end{array}\right]$, that reduces the distinguishing advantage by $2^{-\Omega(n/\log n)}$.*

Sketch of proof. The reduction takes an LWE instance and extends it to a related k-LWE instance for which the additional hint vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ are also known. A major difficulty is to perform this extension while restraining the increase of the noise, 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 [9]. In the latter approach, the hint vectors are chosen independently from a small discrete Gaussian distribution, and then the LWE matrix A is extended to a larger 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 the gadget of Corollary 8. Ignoring several technicalities, the core idea underlying our reduction is that the latter gadget allows us to sample a small matrix $[X_1|X_2]$ with X_2^{-1} also small, which we can then use to transform the given LWE matrix $A^+ \in \mathbb{Z}_q^{m\times n}$ into a taller k-LWE matrix $A'^+ = T \cdot A^+$, using a transformation matrix T of the form

$$T = \left[\frac{I_{m+1}}{-X_2^{-1}X_1}\right].$$

We can accordingly transform the given LWE sample vector $\boldsymbol{b} = A^+\boldsymbol{s} + \boldsymbol{e}$ for matrix A^+ into an LWE sample $\boldsymbol{b'} = T\boldsymbol{b} = A'^+\boldsymbol{s} + T\boldsymbol{e}$ for matrix A'^+ by multiplying by T as well. Since $[X_1|X_2] \cdot T = [X_1|X_2] \cdot A'^+ = 0$, we can use k small rows of X as the k-LWE hints for the new matrix A^+, while, at same time, the smallness of T keeps the transformed noise $\boldsymbol{e}' = T\boldsymbol{e}$ small.

Proof of Theorem 9.

Reduction description. 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 $A' \in \mathbb{Z}_q^{(m+2n)\times 4n}$ and $\boldsymbol{u}' \in \mathbb{Z}_q^{4n}$ independent and uniform, $X \in \mathbb{Z}^{k\times(m+2n)}$ with its ith row $\boldsymbol{x}_i$ independently sampled from $D_{\Lambda^\perp_{-\boldsymbol{u}'}(A'),S}$ for $i \leq k$, and $\boldsymbol{b}' \in \mathbb{T}^{m+1+2n}$ coming from either the 'k-LWE distribution' $\frac{1}{q}U\left(\mathrm{Im}(A'^+)\right) + \nu_\alpha^{m+1+2n}$ if $\boldsymbol{b}$ is from the 'LWE distribution', or the 'k-Uniform distribution' $\frac{1}{q}U\left(\mathrm{Span}_{i\leq k}(\boldsymbol{x}_i^+)^\perp\right)$ 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 $i \leq k$. The reduction is as follows.

1. Sample gadget $\overline{X}_2 \in \mathbb{Z}^{2n\times 2n}$ using the algorithm of Corollary 8 (with parameters n, m_1, m_2, σ_1, σ_2 set to k, n, n, σ, σ' respectively), and sample $\overline{X}_1 \hookleftarrow D_{\mathbb{Z},\sigma}^{2n\times m}$. Define $T = \begin{bmatrix} I_{m+1} \\ -\overline{X}_2^{-1} \cdot (\mathbf{1}|\overline{X}_1) \end{bmatrix} \in \mathbb{Z}^{(m+1+2n)\times(m+1)}$, where $\mathbf{1}$ is the all-1 vector. Let $X \in \mathbb{Z}^{k\times(m+2n)}$ denote the matrix made of the top k rows of $(\overline{X}_1|\overline{X}_2)$.
2. Sample $C^+ \in \mathbb{Z}_q^{(m+1+2n)\times 3n}$ with independent columns uniform orthogonally to $\mathrm{Im}((\mathbf{1}|X))$ modulo q. Let $\boldsymbol{u}_C^t \in \mathbb{Z}_q^{3n}$ denote the top row of C^+, and $C \in \mathbb{Z}_q^{(m+2n)\times 3n}$ denote its remaining $m + 2n$ rows.
3. Compute $\Sigma = \alpha' \cdot I_{m+1+2n} - T \cdot T^t$ and $\sqrt{\Sigma}$ such that $\sqrt{\Sigma} \cdot \sqrt{\Sigma}^t = \Sigma$; if Σ is not positive definite, abort.
4. Compute $A'^+ = (T \cdot A^+ | C^+)$ and $\boldsymbol{b}' = T\boldsymbol{b} + \frac{1}{q}C^+ \cdot \boldsymbol{s}' + \sqrt{\Sigma}\boldsymbol{e}'$, with $\boldsymbol{s}' \hookleftarrow U(\mathbb{Z}_q^{3n})$ and $\boldsymbol{e}' \hookleftarrow \nu_1^{m+1+2n}$. Let $\boldsymbol{u}' = (\boldsymbol{u}\|\boldsymbol{u}_C) \in \mathbb{Z}_q^{4n}$ denote the top row of A'^+.
5. Return $(A', \boldsymbol{u}', X, \boldsymbol{b}')$.

Step 1 aims at building a transformation matrix T that sends A^+ to the left n columns of 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 the gadget of the previous subsection. This ensures that the $\boldsymbol{x}_i$'s are (almost) distributed as independent Gaussian samples from $D_{\mathbb{Z}^n,\sigma} \times D_{\mathbb{Z}^n,\sigma'}$, and that the matrix T is integral with small coefficients. We define $B \in \mathbb{Z}_q^{2n\times n}$ by $[A^+\|B] = TA^+$, so that we have:

$$[\mathbf{1}|\overline{X}_1|\overline{X}_2] \cdot \left[\frac{A^+}{B}\right] = [\mathbf{1}|\overline{X}_1|\overline{X}_2] \cdot \left[\begin{array}{c} I_{m+1} \\ -\overline{X}_2^{-1} \cdot (\mathbf{1}|\overline{X}_1) \end{array}\right] \cdot A^+ = 0 \bmod q.$$

As a consequence, each row of $\left(\overline{X}_1|\overline{X}_2\right)$ belongs to the coset $\Lambda^{\perp}_{-\boldsymbol{u}}(A'')$, where $A'' = [A^t|B^t]^t$.

At this stage, it is tempting to define the k-LWE matrix as A'' and give away the k-LWE hint vectors $\boldsymbol{x}_i \in \Lambda^{\perp}_{-\boldsymbol{u}}(A'')$ making up the matrix X. However, this approach does not quite work: we have extended A by $2n$ rows, but we give only k hint vectors (we cannot output them all, as the bottom rows of $\overline{X}_2$ may not be normally distributed). This creates a difficulty for mapping 'Uniform' to 'k-Uniform' in the reduction.

Step 2 circumvents the above difficulty by sampling extra column vectors $C^+ \in \mathbb{Z}_q^{(m+1+2n)\times 3n}$ that are uniform in the subspace orthogonal to the hint vectors $\boldsymbol{x}_i^+$ modulo q. When the parameters are properly set, the columns of $[T|C^+]$ span the full subspace orthogonal to the $\boldsymbol{x}_i$'s mod q, with overwhelming probability. We finally set $A'^+ = \left[\frac{A^+}{B}\middle| C^+\right]$.

It remains to see how to map 'LWE' to 'k-LWE'. The main problem, when multiplying $\boldsymbol{b}$ by 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 $\alpha^2 T \cdot T^t$. To compensate for that, in Step 3, we add to $T \cdot \boldsymbol{b}$ an independent Gaussian noise with well-chosen covariance $\Sigma = \alpha'^2 \cdot I_{m+1+2n} - \alpha^2 T \cdot T^t$. We set α' large enough to ensure that this symmetric matrix is positive definite. This noise unskewing technique was adapted to discrete Gaussians and used in cryptography in [37].

Reduction analysis. All steps of the reduction can be implemented in polynomial time. Its correctness follows from the following three lemmas. The proofs of these lemmas are given in appendix.

Lemma 10. *The tuple* $(A', \boldsymbol{u}', X)$ *is within statistical distance* $2^{-\Omega(n/\log n)}$ *of the distribution in which* $A' \in \mathbb{Z}_q^{(m+2n)\times 4n}$ *and* $\boldsymbol{u}' \in \mathbb{Z}_q^{4n}$ *are independent and uniform, and the rows of* $X \in \mathbb{Z}^{k \times (m+2n)}$ *are from* $D_{\Lambda^{\perp}_{-\boldsymbol{u}'}(A'),S}$.

Next, we assume that (A'^+, X) is fixed and we 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 11. *The following holds with probability* $1 - 2^{-\Omega(n/\log n)}$ *over the choice of* $\overline{X}_1$ *and* $\overline{X}_2$. *If* $\boldsymbol{b} \in \mathbb{T}^{m+1}$ *is sampled from* $\frac{1}{q}U(\mathrm{Im}A) + \nu_\alpha^{m+1}$, *then* $\boldsymbol{b}' \in \mathbb{T}^{m+1+2n}$ *is within statistical distance* $2^{-\Omega(n)}$ *of* $\frac{1}{q}U\left(\mathrm{Im}A'^+\right) + \nu_{\alpha'}^{m+1+2n}$.

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

Lemma 12. *The following holds with probability* $1 - 2^{-\Omega(n/\log n)}$ *over the choice of* $\overline{X}_1$ *and* $\overline{X}_2$. *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* $\frac{1}{q}U\left(\mathrm{Span}_{i\le k}(\boldsymbol{x}_i^+)^{\perp}\right) + \nu_{\alpha'}^{m+1+2n}$.

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 $2^{-\Omega(n/\log n)}$. □

On the SIS to k-SIS reduction. 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'' in the same way (except that we do not use and add $\boldsymbol{u}$ on top of the matrix A, or equivalently, A^+ is now just identical to A) and then also use the top k rows of $(\overline{X}_1|\overline{X}_2)$ as the k-SIS hints for the new matrix A''. Then, whenever the adversary can output a short vector $\boldsymbol{x_1}\|\boldsymbol{x_2}$ that is orthogonal to A'', we can also output a short vector $(\boldsymbol{x_1} - \boldsymbol{x_2} \cdot \overline{X}_2^{-1}\overline{X}_1)$ which is orthogonal to A. As the rows of $\overline{X}_1$ are distributed as independent Gaussian samples and the adversary is only given its first k rows, it can be shown that, if $\boldsymbol{x_1}\|\boldsymbol{x_2}$ is linearly independent from the k-SIS hints, then the vector $(\boldsymbol{x_1} - \boldsymbol{x_2} \cdot \overline{X}_2^{-1}\overline{X}_1)$ is null with a negligible probability.

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, and finally prove the soundness and confirmation properties of the tracing algorithm.

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 define S as $\mathrm{Diag}(\sigma, \ldots, \sigma, \sigma', \ldots, \sigma') \in \mathbb{R}^{m \times m}$ where $\sigma' > \sigma$ and their respective numbers of iterations are set so that (t, S)-$\mathrm{LWE}_{m+1,\alpha}$ is hard to solve (see Section 3).

`Setup`. The trusted authority generates a master key pair using the algorithm from Lemma 2. 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 algorithm from Lemma 20 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 $\boldsymbol{x}_i^+ = (1 \| \boldsymbol{x}_i) \in \mathbb{Z}^{m+1}$. By Lemma 21 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 [21, 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:

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

As explained in [21], this scheme is semantically secure under chosen plaintext attacks (IND-CPA), under the assumption that $\mathrm{LWE}_{m+1,\alpha}$ is hard to solve.

`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 13. *Let m, n, q, 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`.

[1] The encryption algorithm is often used to encapsulate session keys which are then fed into a highly efficient data encapsulation mechanism to encrypt the data

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 $j \leq k$.

The `Trace` algorithm attempts to gather information about which keys have been used to build the decoder $\mathcal{D}$, by feeding different carefully designed distributions to the 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 the distribution $\mathtt{Enc}(0)$. We define p_∞ as the probability the decoder can decrypt the ciphertexts and p_i the probability the decoder decrypts the signals in Tr_i, for $i \in [0, t]$:

$$p_\infty = \Pr_{\substack{M \hookleftarrow U(\{0,1\}) \\ \boldsymbol{c} \hookleftarrow \mathtt{Enc}(M)}} \left[\mathcal{O}^{\mathcal{D}}(\boldsymbol{c}, M) = 1\right] \text{ and } p_i = \Pr_{\substack{\boldsymbol{c} \hookleftarrow Tr_i \\ M \hookleftarrow U(\{0,1\})}} \left[\mathcal{O}^{\mathcal{D}}\left(\boldsymbol{c} + \left[\frac{M \cdot \lfloor q/2 \rfloor}{\boldsymbol{0}}\right], M\right) = 1\right].$$

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

The confirmation and soundness properties of our schemes are given in Appendix A.3. We now concentrate on a new feature of our scheme: public traceability.

5 Projective sampling and public traceability

We now propose a modification of the scheme of the previous section 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 [18], 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 14 (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

[2] Note that in our context, minimal access is equivalent to standard access: since the plaintext domain size is $\leq \mathcal{P}oly(n)$, the plaintext messages can be tested exhaustively.

secret key and from its projected key are 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 $\boldsymbol{x}_j^{+t} \cdot H_j = \boldsymbol{0}$ 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 the convention that $\cap_{j\leq i}\mathrm{Im}(H_j) = \mathbb{Z}_q^{m+1}$ when $i = 0$. Note that $\cap_{j\leq i}\mathrm{Im}(H_j) \subseteq \mathrm{Span}_{j\leq i}(\boldsymbol{x}_j^+)^{\perp}$.

Theorem 15. *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 uniformy sampled public parameters $(A, \boldsymbol{u})$, the secret key $k = (\boldsymbol{x}_j)_{j\leq i} \hookleftarrow D_i$ and its projected key $\pi_i(k) = (H, (\boldsymbol{h}_j)_{j\leq i})$, 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 $\mathrm{Span}_{j\leq i}(\boldsymbol{x}_j)$ (by Lemma 4). 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 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)-LWE$_{m,\alpha}$, given k, the distributions

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

are indistinguishable. Further, given $k = (\boldsymbol{x}_j)_{j\leq i}$, the projected key $\pi_i(k) = (H, (\boldsymbol{h}_j)_{j\leq i})$ 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, by construction, we have $\mathrm{Im}(A^+) \subseteq \cap_{j\leq i}\mathrm{Im}(H_j) \subseteq (\mathrm{Span}_{j\leq i}(\boldsymbol{x}_j^+))^{\perp}$. Hence:

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

We note that given $\boldsymbol{h}_1, \ldots, \boldsymbol{h}_i$, one can efficiently sample from $U(\cap_{j\leq i}\mathrm{Im}(H_j))$. Therefore, under the hardness of (i, S)-LWE$_{m,\alpha}$, 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. □

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

5.3 Public traceability from projective sampling

In the scheme of Section 4, the tracing key $tk = (T, (\mathcal{U}_i, \boldsymbol{x}_i)_{i \leq N})$ must be kept secret, as it would reveal the secret keys of the users. The tracing signals are samples from $U(\mathrm{Span}_{j \leq i}(\boldsymbol{x}_j^+)^\perp) + \lfloor \nu_{\alpha q} \rceil^{m+1}$, 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 15, 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 $(\boldsymbol{h}_j)_{j \leq N}$ 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 $(\boldsymbol{x}_j)_{j \leq i}$ and all the projected keys $(\boldsymbol{h}_j)_{j \leq N}$, the adversary cannot distinguish $\mathtt{P}_{i,\pi_i(k)}$ from the original signals $U(\mathrm{Im}(A^+)) + \lfloor \nu_{\alpha q} \rceil^{m+1}$. For this purpose, we show how exploit a technique from [22] to simulate $(\boldsymbol{h}_j)_{i < j \leq N}$ from the public information.

Theorem 16. *Set $i \leq t$. Under the (i, S)-$\mathrm{LWE}_{\alpha,m}$ and the $\mathrm{LWE}'_{\alpha,m}$ hardness assumptions, given the secret key $k = (\boldsymbol{x}_j)_{j \leq i}$ and all the projected keys $(H, (\boldsymbol{h}_j)_{j \leq N})$, the following two distributions are indistinguishable*

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

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 $\mathbb{Z}_q^{m+1}$)

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

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 any secret information is required for sampling the projected keys $\boldsymbol{h}_j$'s for $j > i$. In order to prove the indistinguishability of these two games, we adapt a technique due to [22]. This proof of the following lemma is given in Appendix D.

Lemma 17. *Under the $\mathrm{LWE}'_{\alpha,m}$ hardness assumption, the quantity $|\varepsilon_1 - \varepsilon_0|$ is negligible.*

We note that, in **Game**$_1$, the $\boldsymbol{h}_j$'s can be sampled publicly from the available data. Therefore, from Theorem 15, 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 $(H, (\boldsymbol{h}_i)_{i \leq N})$ 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 16, for the particular case of $i = 0$.

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, C. Gentry, S. Halevi, and A. Sahai. Sampling discrete gaussians efficiently and obliviously. Cryptology ePrint Archive, Report 2012/714.
4. M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proc. of STOC*, pages 99–108. ACM, 1996.
5. M. Ajtai. Generating hard instances of the short basis problem. In *Proc. of ICALP*, volume 1644 of *LNCS*, pages 1–9. Springer, 1999.
6. J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. *Theor. Comput. Science*, 48(3):535–553, 2011.
7. O. Billet and D. H. Phan. Efficient Traitor Tracing from Collusion Secure Codes. In Reihaneh Safavi-Naini, editor, *Information Theoretic Security—ICITS 2008*, volume 5155 of *Lecture Notes in Computer Science*, pages 171–182. Springer, 2008.
8. 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. Full version available at `http://crypto.stanford.edu/~dabo/pubs/abstracts/traitors.html`.
9. 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`.
10. 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.
11. D. Boneh and B. Waters. A fully collusion resistant broadcast, trace, and revoke system. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 06*, pages 211–220. ACM Press, October / November 2006.
12. Dan Boneh and Moni Naor. Traitor tracing with constant size ciphertext. In Peng Ning, Paul F. Syverson, and Somesh Jha, editors, *ACM CCS 08*, pages 501–510. ACM Press, October 2008.
13. Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. Classical hardness of learning with errors. In *STOC*, pages 575–584. ACM, 2013.
14. H. Chabanne, D. H. Phan, and D. Pointcheval. Public traceability in traitor tracing schemes. In *Proc. of EUROCRYPT*, volume 3494 of *LNCS*, pages 542–558. Springer, 2005.
15. Hervé Chabanne, Duong Hieu Phan, and David Pointcheval. Public traceability in traitor tracing schemes. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 542–558. Springer, May 2005.
16. B. Chor, A. Fiat, and M. Naor. Tracing traitors. In *Proc. of CRYPTO*, volume 839 of *LNCS*, pages 257–270. Springer, 1994.
17. B. Chor, A. Fiat, M. Naor, and B. Pinkas. Tracing traitors. *IEEE Trans. Inf. Th.*, 46(3):893–910, 2000.
18. Ronald Cramer and Victor 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.
19. 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.
20. S. Garg, C. Gentry, and S. Halevi. Candidate multilinear maps from ideal lattices. In *EUROCRYPT*, volume 7881 of *LNCS*, pages 1–17, 2013.
21. 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`.
22. 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.
23. A. Kiayias and S. Pehlivanglu. *Encryption For Digital Content.* Springer, 2010.
24. 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.
25. 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.
26. A. Kiayias and M. Yung. Breaking and repairing asymmetric public-key traitor tracing. In *Digital Rights Management Workshop*, pages 32–50, 2002.
27. 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.
28. 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.

29. K. Kurosawa and Y. Desmedt. Optimum traitor tracing and asymmetric schemes. In *Proc. of EUROCRYPT*, LNCS, pages 145–157. Springer, 1998.
30. 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.
31. 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.
32. D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput*, 37(1):267–302, 2007.
33. 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.
34. Moni Naor and Benny Pinkas. Efficient trace and revoke schemes. In Yair Frankel, editor, *FC 2000*, volume 1962 of *LNCS*, pages 1–20. Springer, February 2000.
35. 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.
36. C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *Proc. of STOC*, pages 333–342. ACM, 2009.
37. C. Peikert. An efficient and parallel Gaussian sampler for lattices. In *Proc. of CRYPTO*, volume 6223 of *LNCS*, pages 80–97. Springer, 2010.
38. C. Peikert, A. Shelat, and A. Smith. Lower bounds for collusion-secure fingerprinting. In *Proc. of SODA*, pages 472–479, 2003.
39. B. Pfitzmann. Trials of traced traitors. In *Information Hiding*, volume 1174 of *LNCS*, pages 49–64. Springer, 1996.
40. B. Pfitzmann and M. Waidner. Asymmetric fingerprinting for larger collusions. In *ACM CCS 97*, pages 151–160. ACM Press, April 1997.
41. 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.
42. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proc. of STOC*, pages 84–93. ACM, 2005.
43. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009.
44. O. Regev. The learning with errors problem, 2010. Invited survey in CCC 2010, available at `http://www.cims.nyu.edu/~regev/`.
45. 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.
46. Thomas 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.
47. 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.
48. 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.
49. G. Tardos. Optimal probabilistic fingerprint codes. *J. ACM*, 55(2), 2008.
50. 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.

A Traitor Tracing

A.1 A short overview

COMBINATORIAL SCHEMES VERSUS ALGEBRAIC SCHEMES. There are two main approaches for devising a traitor tracing encryption scheme. Many constructions are combinatorial in nature (see [16, 47, 17, 45, 41, 7, 12], 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 [49]: 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 [38, 49], leaving little hope of significant improvements.

An alternative approach was initiated by Kurosawa and Desmedt in [29] (whose construction was shown insecure in [48]), and by Boneh and Franklin [8]: 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. We will concentrate on the algebraic approach in this paper. Prior to this work, all known algebraic traitor tracing schemes relied on variants of the Discrete Logarithm Problem: For instance, the earlier constructions (including [29, 8, 27, 30]) rely on the assumed hardness of the Decision Diffie Hellman problem (DDH), whereas others (including [14, 10, 11, 1, 19]) 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 traitors 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 [39], a commitment protocol in [40], an oblivious polynomial evaluation in [50, 28, 26].

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 traitors 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 [41, 11] (some schemes, such as [15, 7, 12], partially achieve: some parts of the tracing procedure can be run publicly). The scheme [41] is generic, based on IPP-codes, and is thus quite impractical. The Boneh-Waters scheme [11] 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.

A.2 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 $(\mathcal{U}_i)_{i \leq N}$ 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 $(sk_i)_{i \leq N}$, 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 $i \leq N$.
- 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

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

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 [8]. 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 $1/|\mathcal{P}|$:

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

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 [27]). 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 [25, 24].

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 $k \leq t$, and must satisfy the following two properties:

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

A.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.

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

Proof. Wlog we may assume that the traitors in the coalition know all the secret keys $sk_1, \dots, sk_t$. The hardness of (t,S)-LWE$_{m+1,\alpha}$ implies that the distributions $\texttt{Enc}(0)$ 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 $\le \frac{1}{2}$. As $p_t - p_0 > \frac{\varepsilon}{2}$ and $|\widetilde{p}_i - p_i| \le \frac{\varepsilon}{8}$ for all i, we must have $\widetilde{p}_t - \widetilde{p}_0 > \frac{\varepsilon}{4} \ge \frac{\widetilde{\varepsilon}}{8}$, with probability exponentially close to 1. As a consequence, there must exist $i \le t$ 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 3 several times.

Theorem 19. *Assume that decoder $\mathcal{D}$ was built using $\{sk_{i_j}\}_{j\le k}$. Under the parameter assumptions of Theorem 3 with Theorem 3's (k,n) set to $(t{+}1, n{+}t{+}1)$, and the computational assumption that $(t{+}1,S)$-LWE$_{m+1,\alpha}$ is hard: if algorithm* `Trace` *returns "User $\mathcal{U}_{i_0}$ is guilty", then $i_0 \in \{i_j\}_{j\le k}$.*

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

$\texttt{Game}_0$. The game consists of three steps, as follows:

- $\texttt{Initialize}_0$: 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 $i \le t+1$.
- $\texttt{Input}_0$: Send $A^+ = (\boldsymbol{u}^t \| A)$ and $(\boldsymbol{x}_i)_{i\le t+1, i\ne i_0}$ to $\mathcal{A}$.
- $\texttt{Challenge}_0$: Sample $b \hookleftarrow U(\{0,1\})$. Send to $\mathcal{A}$ arbitrarily many samples from $U\left(\mathrm{Span}_{i\le i_0-1+b}(\boldsymbol{x}_i^+)^\perp\right) + \lfloor \nu_{\alpha q} \rceil^{m+1}$.

We say that $\mathcal{A}$ wins $\texttt{Game}_0$ 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 $(\boldsymbol{c} + (M|\boldsymbol{0}^t)^t, M)$ as input to $\mathcal{O}^{\mathcal{D}}$, where $\boldsymbol{c}$ is any sample from $\texttt{Challenge}_0$.We now introduce two variations of $\texttt{Game}_0$, which differ in the Initialize and Challenge steps.

$\texttt{Game}_1$. 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 $i \le t+1$, and $\boldsymbol{b}_j^+ \hookleftarrow U(\mathrm{Span}_{i<i_0}(\boldsymbol{x}_i^+)^\perp)$ for $j \le t - i_0 + 2$.
- $\texttt{Input}_1$: Send $A^+ = (\boldsymbol{u}^t \| A)$ and $(\boldsymbol{x}_i)_{i\le t+1, i\ne i_0}$ 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:

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

As in $\texttt{Game}_0$, algorithm $\mathcal{A}$ wins $\texttt{Game}_1$ if it guesses b with non-negligible advantage.

$\texttt{Game}_1'$ is as $\texttt{Game}_1$, except that if $b=0$ in the challenge step, then the samples sent to $\mathcal{A}$ are from the distribution $U\left(\mathrm{Span}_{i\le i_0}(\boldsymbol{x}_i^+)^\perp\right) + \lfloor \nu_{\alpha q} \rceil^{m+1}$. (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 $\texttt{Game}_0$, $\texttt{Game}_1$ and $\texttt{Game}_1'$ are identical (only the distributions of the Challenge steps vary). By the triangle inequality, if $\mathcal{A}$ wins $\texttt{Game}_0$ with some non-negligible advantage, then it may be used to win either $\texttt{Game}_1$ or $\texttt{Game}_1'$ 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 $\texttt{Game}_1$ 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 $(A, \boldsymbol{u}, (\boldsymbol{x}_i)_{i \le t+1})$, and that we aim at distinguishing between the following distributions over $\mathbb{Z}_q^{m+1}$:

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

To solve this problem instance, we sample $\boldsymbol{b}_j$ for $j \le t - i_0 + 2$ 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 $m \ge t+n$, these $(t - i_0 + 2)$ vectors are linearly independent and none of them belongs to $\mathrm{Span}_{i_0 \le i \le t+1}(\boldsymbol{x}_i^+)^\perp$, with probability $\ge 1 - 2^{-\Omega(n)}$. In that case, the transformation maps $U\left(\mathrm{Span}_{i \le t+1}(\boldsymbol{x}_i^+)^\perp\right) + \nu_{\alpha q}^{m+1}$ 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 $U(\mathrm{Im}[A^+|\boldsymbol{b}_1^+|\dots|\boldsymbol{b}_{t-i_0+2}^+]) + \nu_{\alpha q}^{m+1}$. 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 $\texttt{Game}_1'$ 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: $\mathrm{Im}([A^+|\boldsymbol{b}_1|\dots|\boldsymbol{b}_{t-i_0+2}]) \subseteq \mathrm{Span}_{i<i_0}(\boldsymbol{x}_i^+)^\perp$. In the second situation, no such inclusion holds. The purpose of the sequence of games below is to map $\texttt{Game}_1'$ to recover such an inclusion setting.

Let us define $\texttt{Game}_2$ as being the same as $\texttt{Game}_1'$, 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
$$A' = [A|\boldsymbol{b}_1|\dots|\boldsymbol{b}_{t-i_0+2}] \quad \text{and} \quad \boldsymbol{u}' = (\boldsymbol{u}\|v_1\|\dots\|v_{t-i_0+2}).$$

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 3 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 3 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 $\texttt{Game}_2$ with non-negligible advantage.

Now, consider $\texttt{Game}_3$, which differs from $\texttt{Game}_2$ 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 $U\left(\mathrm{Span}_{i<i_0}(\boldsymbol{x}_i^+)^\perp\right) + \lfloor \nu_{\alpha q} \rceil^{m+1}$ and $U(\mathrm{Im}[A^+|\boldsymbol{b}_1|\dots|\boldsymbol{b}_{t-i_0+2}]) + \lfloor \nu_{\alpha q} \rceil^{m+1}$, this modification does

not alter the winning probability of $\mathcal{A}$: algorithm $\mathcal{A}$ also wins $\texttt{Game}_3$ with non-negligible advantage. Now, we again use Theorem 3 twice, but with $(\boldsymbol{x}_i)_{i \leq i_0}$: 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 $\texttt{Game}_4$ with non-negligible advantage, where $\texttt{Game}_4$ differs from $\texttt{Game}_3$ only in its first step, as follows.

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

The situation we are in now is very similar to that in the first situation, where $\mathcal{A}$ was supposed to win $\texttt{Game}_1$. The arguments used in the first situation readily carry over here (up to replacing $\mathrm{Span}_{i<i_0}\boldsymbol{x}_i^+$ and $\mathrm{Span}_{i \geq i_0}\boldsymbol{x}_i^+$ by $\mathrm{Span}_{i \leq i_0}\boldsymbol{x}_i^+$ and $\mathrm{Span}_{i>i_0}\boldsymbol{x}_i^+$, respectively). □

B Basic results on lattices

Gentry et al. [21] gave an algorithm to sample from $D_{L,S,\boldsymbol{c}}$

Lemma 20 ([13, 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 $\sqrt{\ln(2n+4)/\pi} \cdot \max_i \|S^{-t}\boldsymbol{b}_i\| \leq 1$, 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 21 (Adapted from [3, 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 $\sigma_m(S) \geq \eta_\varepsilon(L)$ with $\varepsilon \in (0, 1/2)$, we have $\Pr_{\boldsymbol{b} \hookleftarrow D_{L,S,\boldsymbol{c}}}[\|\boldsymbol{b} - \boldsymbol{c}\| \geq \sigma_1(S) \cdot \sqrt{n}] \leq 2^{-n+2}$.*

Lemma 22 (Adapted from [32, 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 $\sigma_m(S) \geq \eta_\varepsilon(L)$ 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 23 (Special case of [37, 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 with $1 \geq \eta_\varepsilon(S_1^{-1}\Lambda_1)$ and $1 \geq \eta_\varepsilon(\sqrt{(S_1S_1^t)^{-1} + (S_2S_2^t)^{-1}} \cdot \Lambda_2)$ for some $\varepsilon \in (0, 1/2)$. If $\boldsymbol{x}_2 \hookleftarrow D_{\Lambda_2,S_2,\boldsymbol{0}}$ 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 24 ([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 $\Delta(X^t \cdot D_{\mathbb{Z}^m,S,\boldsymbol{c}}, D_{\mathbb{Z}^n,SX,S^t\boldsymbol{c}}) \leq 2\varepsilon$.*

Lemma 25 ([3, Le. 8]). *Let $n \geq 1$, $m \geq 2n$, and $\sigma \geq C \cdot \sqrt{n}$ for some absolute constant C. Let $X \hookleftarrow D_{\mathbb{Z},\sigma}^{m \times n}$. Then, except with probability $2^{-\Omega(m)}$, we have $\sigma_n(X) \geq \Omega(\sigma\sqrt{m})$.*

C Missing proofs of Section 3

Proof of Lemma 6. We apply Lemma 24 with S invertible chosen such that $(SX)(SX)^t = \sigma_2^2 I_m$ for some $\sigma_2 > \sigma_1$, thus obtaining an unskewed Gaussian distribution $D_{\mathbb{Z}^n,\sigma_2}$. The scaling σ_2 is chosen sufficiently large so that the assumptions of Lemmas 24 and 25 hold.

We first sample X from $D_{\mathbb{Z},\sigma_1}^{m \times n}$, using Lemma 20. By Lemma 24 (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 $D_{\mathbb{Z}^m,S,\boldsymbol{a}}$, using Lemma 20 again, for some invertible matrix $S \in \mathbb{R}^{m \times m}$

and vector $\boldsymbol{a} \in \mathbb{Z}^m$ chosen as described below. Finally, we set $\boldsymbol{x} = \boldsymbol{c} + X^t \cdot \boldsymbol{r}$. If the assumptions of Lemma 24 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 $D_{\mathbb{Z}^n, SX, X^t\boldsymbol{a}+\boldsymbol{c}}$.

We set $\boldsymbol{a} \in \mathbb{Z}^m$ so that $X^t\boldsymbol{a} + \boldsymbol{c} = \boldsymbol{0}$ (this is possible, as the row $\mathbb{Z}$-span of X is $\mathbb{Z}^n$). Now, we build S using the singular value decomposition $X = U_X \cdot \mathrm{Diag}((\sigma_i(X))_{i \leq n}) \cdot V_X$, where $U_X \in \mathbb{R}^{m \times n}$ and $V_X \in \mathbb{R}^{n \times n}$ are orthogonal matrices. We define $S = U_S \cdot \mathrm{Diag}((s_i)_{i \leq m}) \cdot V_S$ as follows: we set $U_S^t = \begin{bmatrix} V_X & \boldsymbol{0} \\ \boldsymbol{0} & I_{m-n} \end{bmatrix}$ and $V_S^t = [U_X | U_X^\perp]$, where $U_X^\perp$ is an orthonormal basis for the orthogonal of $U_X \cdot \mathbb{R}^n$; we also set $s_i = \sigma_2 / \sigma_i(X)$ for $i \leq n$ and $s_i = \sigma_n(S)$ for $i > n$. This leads to $SX = \sigma_2 \cdot I_n$.

To check that the assumptions of Lemma 24 are satisfied, note that the smallest singular value of S is $\sigma_m(S) = s_1 = \sigma_2/\sigma_1(X)$. Hence the assumption $\sigma_m(S) \geq 10n\sigma_1 \log^{3/2}(nm\sigma_1/\varepsilon)$ is satisfied if $\sigma_2 \geq \sigma_1(X) \cdot 10n\sigma_1 \log^{3/2}(nm\sigma_1/\varepsilon)$. The latter holds by the choice of σ_2, using the fact that $\sigma_1(X) \leq \|X\| \leq \sqrt{m} \cdot \sigma_1$. The second inequality holds with probability $\geq 1 - n2^{-m+2}$, using the union bound and Lemma 21.

Finally, the bound on $\|\boldsymbol{r}\|$ follows from Lemma 21 and the facts that $\sigma_1(S) = \sigma_2/\sigma_n(X)$ and $\sigma_n(X) \geq \Omega(\sigma\sqrt{m})$ except with probability $2^{-\Omega(m)}$, by Lemma 25. □

Proof of Lemma 10. Let D_0 denote the desired distribution for $(A', \boldsymbol{u}', X)$. We first apply Theorem 3 (with the theorem parameters $m, n, k, \sigma_1(S), \sigma_m(S)$ having the values $m + 2n$, $3n$, $n/(c \log n)$, 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: $\boldsymbol{u}' \in \mathbb{Z}_q^{3n}$ is sampled uniformly, $X \in \mathbb{Z}^{k \times (m+2n)}$ has its ith row $\boldsymbol{x}_i$ independently sampled from $D_{\mathbb{Z}^{m+2n}, S}$, and $A' \in \mathbb{Z}_q^{(m+2n) \times 3n}$ 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.

Next, let $A' = \left(\begin{matrix} A \\ B \end{matrix} \middle| C \right)$, where $A \in \mathbb{Z}_q^{m \times n}$, $B \in \mathbb{Z}_q^{2n \times n}$ and $C \in \mathbb{Z}_q^{(m+2n) \times 3n}$. Note that in the distribution D_1, *all* of A' is chosen uniformly from the set of solutions to $X \cdot A' = U' \bmod q$ (where $U' \in \mathbb{Z}_q^{k \times 3n}$ consists of k copies of $\boldsymbol{u}'^t$). We now show that D_1 is within statistical distance $2^{-\Omega(n)}$ to the distribution D_2 that is defined as D_1, except that in D_2, the submatrix $A \in \mathbb{Z}_q^{m \times n}$ is chosen independently uniformly at random, and then B, C are chosen uniformly from the set of solutions to $X \cdot A' = U' \bmod q$. The distribution of $(C, \boldsymbol{u}', X)$ is the same in D_1 and D_2, by definition. The condition on (A, B) in D_1 is $X_1 \cdot A + X_2 \cdot B = U \bmod q$, where $X_1 \in \mathbb{Z}^{k \times m}$ and $X_2 \in \mathbb{Z}^{k \times 2n}$ are the left and right submatrices of X, respectively, and $U \in \mathbb{Z}_q^{k \times n}$ consists of the n left columns of U'. If X_2 has full rank k over $\mathbb{Z}_q$, then for every choice of $A \in \mathbb{Z}_q^{m \times n}$, the latter equation has the same number of solutions for $B \in \mathbb{Z}_q^{2n \times n}$ (namely $q^{(2n-k) \cdot n}$). Hence, conditioned on X_2 having rank k, the distribution of (A, B) is the same in D_1 and D_2. Therefore, the statistical distance $\Delta(D_1, D_2)$ is $2^{-\Omega(n)}$ if the probability that X_2 has rank k in D_1 is $2^{-\Omega(n)}$. The latter holds by Lemma 4 and our choice of parameters.

Finally, let D_3 denote the distribution of $(A', \boldsymbol{u}', X)$ in the reduction. We show below that $\Delta(D_2, D_3) \leq 2^{-\Omega(n/\log n)}$, which completes the proof.

First, we consider the distribution of X. By Corollary 8, we have that, in distribution D_3, the last $2n$ columns of X are within statistical distance $\varepsilon_1 = 2^{-\Omega(n/\log n)}$ of $D_{\mathbb{Z},\sigma}^{k \times n} \times D_{\mathbb{Z},\sigma'}^{k \times n}$. Since the first m columns of X are independently distributed as $D_{\mathbb{Z},\sigma}^{k \times m}$ in both D_2 and D_3, it follows that the distribution of X in D_3 is within statistical distance $\varepsilon_1 = 2^{-\Omega(n/\log n)}$ of its distribution $D_{\mathbb{Z}^{m+2n}, S}$ in D_2.

Next, we consider the distribution of A' given some fixed $(\boldsymbol{u}', X)$. Observe that the only difference between these conditional distributions in D_2 and D_3 is that in D_3, matrix B is defined as the unique solution to $(\boldsymbol{1}|\overline{X_1}) \cdot (\boldsymbol{u}^t \| A) + \overline{X_2} \cdot B = 0 \bmod q$, whereas in D_2, matrix B is chosen uniformly among the solutions to $(\boldsymbol{1}|X_1) \cdot (\boldsymbol{u}^t \| A) + X_2 \cdot B = 0 \bmod q$, where X_1, X_2 are the top k rows of $\overline{X}_1, \overline{X}_2$, respectively. We show that these conditional distributions are within statistical distance $\varepsilon_2 = 2^{-\Omega(n)}$, which immediately implies that $\Delta(D_2, D_3) \leq \varepsilon_1 + \varepsilon_2 = 2^{-\Omega(n/\log n)}$, as required.

To see this, let X_1', X_2' denote the bottom $2n-k$ rows of $\overline{X}_1, \overline{X}_2$, respectively. Fix $X_1, X_2, X_2', \boldsymbol{u}$, A, with A such that $\eta_{2^{-n}}(\Lambda^\perp(A)) = O(\sqrt{n \log m}) \cdot q^{\frac{n}{m}}$. By Lemma 1, this condition holds with probability $1-2^{-\Omega(n)}$ over the uniform choice of A. Let B^* denote any solution to $(\mathbf{1}|X_1)\cdot(\boldsymbol{u}^t\|A)+X_2\cdot B = 0 \bmod q$. Let $p(B^*)$ denote the probability that $B = B^*$ in distribution D_3, conditioned on $X_1, X_2, X_2', \boldsymbol{u}, A$. We show that $p(B^*)$ is of the form $(1+\varepsilon_{B^*})\cdot K$ for any such B^*, for $\varepsilon_{B^*} \leq 2^{-\Omega(n)}$ and some normalization constant K independent of B^*. From this it follows immediately that, in D_3, the conditional distribution of B is within distance $2^{-\Omega(n)}$ of the uniform distribution on the set of solutions to $(\mathbf{1}|X_1)\cdot(\boldsymbol{u}^t\|A) + X_2 \cdot B = 0 \bmod q$, which is the conditional distribution of B in D_2, and our claim follows immediately. The probability $p(B^*)$ is the probability that $X_1' \cdot A + X_2' \cdot B = U \bmod q$, conditioned on $X_1, X_2, X_2', \boldsymbol{u}, A$. Let $\boldsymbol{x}_{1,i}' \in \mathbb{Z}^m$ and $\boldsymbol{x}_{2,i}' \in \mathbb{Z}^{2n}$ denote the ith rows of X_1' and X_2', respectively, for $i \leq 2n-k$. Observe that the set of solutions for $\boldsymbol{x}_{1,i}' \in \mathbb{Z}^m$ to $\boldsymbol{x}_{1,i}'^t \cdot A + \boldsymbol{x}_{2,i}'^t \cdot B^* = -\boldsymbol{u}^t \bmod q$ is the coset $\Lambda^\perp_{-\boldsymbol{u}-\boldsymbol{x}_{2,i}' \cdot B^*}(A)$ and, since $\boldsymbol{x}_{1,i}'$ is independently distributed as $D_{\mathbb{Z}^m,\sigma}$ for each i, it follows that

$$p(B^*) = \prod_{i \leq 2n-k} D_{\mathbb{Z}^m,\sigma}(\Lambda^\perp_{-\boldsymbol{u}-\boldsymbol{x}_{2,i}'^t \cdot B^*}(A)) = \prod_{i \leq 2n-k} \rho_{\sigma,\boldsymbol{c}_i}(\Lambda^\perp(A))/\rho_\sigma(\mathbb{Z}^m),$$

for some $\boldsymbol{c}_i \in \mathbb{Z}_q^m$ such that $\boldsymbol{c}_i^t \cdot A = \boldsymbol{u}^t + \boldsymbol{x}_{2,i}'^t \cdot B^* \bmod q$. By Lemma 22, using the choice of $\sigma \geq \eta_{2^{-n}}(\Lambda^\perp(A)) = O(\sqrt{n \log m}) \cdot q^{\frac{n}{m}}$, we have $\rho_{\sigma,\boldsymbol{c}_i}(\Lambda^\perp(A)) = (1 + \varepsilon_{B^*}') \cdot \rho_\sigma(\Lambda^\perp(A))$ for some $\varepsilon_{B^*}' \leq 2^{-\Omega(n)}$. It follows that $p(B^*) \sim 1 + \varepsilon_{B^*}$ for some $\varepsilon_{B^*} \leq n2^{-\Omega(n)}$. □

Proof of Lemma 11. In our proof, we need to use a bound on the probability that a collection of vectors $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_{d+w}$ uniformly and independently sampled from a linear subspace X of dimension d over $\mathbb{Z}_q$, spans X. This is given by the following proposition.

Proposition 26. *Let $d, w, q > 0$ with q prime. Let X denote a d-dimensional $\mathbb{Z}_q$-linear space. Let $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_{d+w} \in X$ be independently sampled from $U(X)$. Then* $\mathrm{Span}_{i \leq d+w}(\boldsymbol{t}_i) = X$*, with probability* $\geq 1 - 2^{d+w}/q^{w+1}$.

Proof. For $i \leq d+w$, let χ_i denote the Bernoulli random variable that is 0 if $\boldsymbol{t}_i \in \mathrm{Span}_{j<i}(\boldsymbol{t}_j)$ and 1 else. Let r_i denote the rank of $\mathrm{Span}_{j \leq i}(\boldsymbol{t}_j)$. Since $r_i = r_{i-1} + \chi_i$, we have $r_{d+w} = \sum_{i=1}^{d+w} \chi_i$. Let S denote the set of binary vectors of length $d+w$ and weight $< d$. Then it suffices to bound the probability that $\boldsymbol{\chi} = (\chi_1, \ldots, \chi_{d+w}) \in S$. To do so, let $\boldsymbol{\chi}' = (\chi_1', \ldots, \chi_{d+w}') \in \{0,1\}^{d+w}$ denote any fixed vector in S. Note that for any $i \leq d+w$, we have $\Pr[\chi_i = 0 | \chi_j = \chi_j' \text{ for } j < i] = q^{\sum_{j<i} \chi_j'}/q^d \leq 1/q$, since $\boldsymbol{\chi}' \in S$. It follows that $\Pr[\boldsymbol{\chi} = \boldsymbol{\chi}'] \leq 1/q^z$, where z denotes the number of zero entries in $\boldsymbol{\chi}'$. Since the weight of $\boldsymbol{\chi}'$ is $< d$, we have $z > d+w-d = w$, so $\Pr[\boldsymbol{\chi} = \boldsymbol{\chi}'] \leq 1/q^{w+1}$. Taking a union bound over all $\boldsymbol{\chi}' \in S$, and using $|S| \leq 2^{d+w}$ completes the proof. □

We now prove the lemma. 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

$$\boldsymbol{b}' = T \cdot \boldsymbol{b} + \frac{1}{q}C^+ \cdot \boldsymbol{s}' + \sqrt{\Sigma}\boldsymbol{e}' = \frac{1}{q} \cdot TA^+ \cdot \boldsymbol{s} + \frac{1}{q}C^+ \cdot \boldsymbol{s}' + T \cdot \boldsymbol{e} + \sqrt{\Sigma}\boldsymbol{e}' = \frac{1}{q}A'^+ \cdot \begin{bmatrix} \boldsymbol{s} \\ \boldsymbol{s}' \end{bmatrix} + T \cdot \boldsymbol{e} + \sqrt{\Sigma}\boldsymbol{e}'.$$

Now, since $\boldsymbol{s}$ and $\boldsymbol{s}'$ are uniform and independent, we have $\frac{1}{q}A'^+ \cdot [\boldsymbol{s}\|\boldsymbol{s}']$ is uniformly distributed in $\mathrm{Im}(A'^+)$. Moreover, the vector $T \cdot \boldsymbol{e}$ is normally distributed with covariance matrix $\alpha^2 \cdot TT^t$, while $\sqrt{\Sigma}\boldsymbol{e}'$ is independent and normally distributed with covariance matrix $\Sigma = \alpha'^2 I_{m+1+2n} - \alpha^2 TT^t$ (we show below that Σ is indeed a valid covariance matrix, i.e., is positive definite, so that $\sqrt{\Sigma}$

exists, except with probability $2^{-\Omega(n/\log n)}$). Therefore, the vector $T \cdot \boldsymbol{e} + \sqrt{\Sigma}\boldsymbol{e'}$ has distribution $\nu_{\alpha'}^{m+1+2n}$, as required.

It remains to show that $\Sigma = \alpha'^2 I_{m+1+2n} - \alpha^2 TT^t$ is a positive definite matrix, with overwhelming probability over the choice of $\overline{X}_1$ and $\overline{X}_2$. By definition, the singular values of Σ are of the form $\alpha'^2 - \alpha^2 \sigma_i(T)^2$, where the $\sigma_i(T)$'s are the singular values of T. It therefore suffices to show that $\alpha'^2 > \alpha^2 \sigma_1(T)^2$, where $\sigma_1(T)$ is the largest singular value of T. We have $\sigma_1(T) \leq \sqrt{m+1}\|T\|$ (by Schwarz's inequality). Each column of T has norm $\leq \sqrt{1 + (m+1)\|\overline{X}_2^{-1}\|^2 t^2}$, where t denotes the maximum column norm of the matrix $(\boldsymbol{1}|\overline{X}_1)$. Since the columns of $\overline{X}_1$ are sampled from $D_{\mathbb{Z}^{2n},\sigma}$, we have by Lemma 21 that $t \leq \sigma \cdot \sqrt{2n}$, and by Corollary 8 that $\|\overline{X}_2^{-1}\| = O(\sigma' n)$, with both bounds holding with probability $\geq 1 - 2^{-\Omega(n/\log n)}$. It follows that $\sigma_1(T) = O(mn^{3/2}\sigma\sigma')$, and hence the assumption that $\alpha' = \Omega(mn^{3/2}\sigma\sigma'\alpha)$ allows us to complete the proof. □

Proof of Lemma 12. 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

$$\boldsymbol{b}' = \frac{1}{q}T \cdot \boldsymbol{y} + \frac{1}{q}C^+ \cdot \boldsymbol{s}' + T \cdot \boldsymbol{e} + \sqrt{\Sigma}\boldsymbol{e}' = \frac{1}{q}[T|C^+] \cdot \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{s}' \end{bmatrix} + T \cdot \boldsymbol{e} + \sqrt{\Sigma}\boldsymbol{e}'.$$

Now, since $\boldsymbol{y}$ and $\boldsymbol{s}'$ are uniform and independent, we have that $\frac{1}{q}[T|C^+] \cdot [\boldsymbol{y}\|\boldsymbol{s}']$ is uniform in $\mathrm{Im}([T|C^+])$.

By construction of T and C, we have that $\mathrm{Im}([T|C^+])$ is a subspace of $X^\perp = \left(\mathrm{Span}_{i\leq k}(\boldsymbol{x}_i^+)^\perp\right)$. We claim that in fact $\mathrm{Im}([T|C^+]) = X^\perp$, except with probability $2^{-\Omega(n/\log n)}$ over the choice of the $\boldsymbol{x}_i$'s and C^+. Indeed, by Lemmas 10 and 4, with probability $\geq 1 - 2^{-\Omega(n/\log n)}$, the vectors $\boldsymbol{x}_1^+, \ldots, \boldsymbol{x}_k^+$ are linearly independent over $\mathbb{Z}_q$ and hence the subspace $X^\perp$ has dimension $m+1+2n-k$. Now, the $m+1$ columns of T are linearly independent. Hence, it suffices to show that the $3n$ projections of the columns of C^+ on the orthogonal complement of $\mathrm{Im}(T) \subseteq X^\perp$ span that $(2n-k)$-dimensional space. As these projections are uniform, we can apply Proposition 26, which tells us this is the case with probability $\geq 1 - 2^{3n}/q^{n+k+1} \geq 1 - 2^{-\Omega(n)}$.

We have showed that $\frac{1}{q}[T|C^+] \cdot [\boldsymbol{y}\|\boldsymbol{s}']$ is within statistical distance $\leq 2^{-\Omega(n)}$ of $\frac{1}{q}U(X^\perp)$, with probability $\geq 1 - 2^{-\Omega(n/\log n)}$ over the choice of X. As shown in Lemma 11, we also have that the noise term $T \cdot \boldsymbol{e} + \sqrt{\Sigma}\boldsymbol{e'}$ is within statistical distance $2^{-\Omega(n)}$ of the distribution ν_α^{m+1+2n}, as required. □

D Missing proof of Section 5

Proof of Lemma 17. Our aim is to reduce $\mathrm{LWE}'_{\alpha,m+1}$ to distinguishing $\mathbf{Game}_1$ and $\mathbf{Game}_0$. 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 $\mathbf{Game}_0$ and $\mathbf{Game}_1$ to decide whether the $\boldsymbol{e}_j$'s are Gaussian or uniform. We simulate $\mathbf{Game}_1$ and $\mathbf{Game}_0$ 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 [22, 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 $\mathbf{Game}_0$ and $\mathbf{Game}_1$.
- 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 $S_2 = \sqrt{SS^t - \alpha^2 q^2 I_m}$ (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 $(\boldsymbol{e}_j^t + \boldsymbol{e}'^t_j) \cdot A = \boldsymbol{z}_j^t \cdot A = \boldsymbol{c}^t \cdot A = \boldsymbol{u}^t$.
  - Set $\boldsymbol{h}_j^t = -\boldsymbol{z}_j^t \cdot H$. Note that $\boldsymbol{h}_j^t = -(\boldsymbol{e}_j^t + \boldsymbol{e}'^t_j) \cdot H$.
- 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 $\boldsymbol{z}_j = \boldsymbol{y}_j + \boldsymbol{e}'_j = B \cdot s_j + (\boldsymbol{e}_j + \boldsymbol{e}'_j)$. 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 23, whose assumptions are satisfied (thanks to the second lower bound on σ') and to Lemma 1; consequently, the residual distribution of $\boldsymbol{e}_j + \boldsymbol{e}'_j$ is negligibly close to $\boldsymbol{c} + D_{\Lambda^{\perp}(A),S,-\boldsymbol{c}}$, 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 $\mathbf{Game}_0$, 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 $\mathbf{Game}_1$, up to negligible statistical distance.

This completes the proof of the lemma. □