A Lattice-Based Traitor Tracing Scheme with Public Traceability

Abstract. A traitor tracing scheme is a multi-receiver encryption scheme with a tracing algorithm aimed at deterring malicious coalitions from building pirate decryption devices. All existing algebraic traitor tracing schemes exploit the hardness of variants of the Discrete Logarithm Problem. In this work, we present the first algebraic construction of a traitor tracing encryption scheme whose security relies on the worst-case hardness of standard lattice problems. The scheme is public-key and provably resists chosen plaintext attacks. Further, it achieves public traceability, i.e., allows the authority to delegate the tracing capability to "untrusted" parties. Our construction is based on the introduction of new problems and techniques:

- We introduce the k-LWE *problem*, which is a Learning With Errors variant of the k-SIS problem from Boneh and Freeman [PKC'11]. The Boneh-Freeman reduction from SIS to k-SIS presents 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. The security of our scheme relies on the hardness of k-LWE.
- We introduce a new *oblivious sampling in a subspace* technique that enables public sampling of a signal from a subspace contained in a secret vector space. This technique allows us to achieve public traceability.

In term of efficiency, although dealing with multi-receiver setting and tracing traitors, our proposed scheme remains as efficient as standard LWE public key encryption, for a large number of traitors k.

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

1 Introduction

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. [18], much work has been devoted to devising efficient and secure traitor tracing schemes. We refer to [24] for an introduction to this rich topic.

Combinatorial schemes versus algebraic schemes. There are two main approaches for devising a traitor tracing encryption scheme. Many constructions are combinatorial in nature (see [18, 50, 19, 49, 45, 8, 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 [52]: To resist coalitions of up to t malicious users among N users, the code length is $\ell = O(t^2 \log N)$. Lower bounds with the same dependence with respect to t have been given in [41, 52], leaving little hope of significant improvements.

An alternative approach was initiated by Kurosawa and Desmedt in [31] (whose construction was shown insecure in [51]), and by Boneh and Franklin [10]: 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. Prior to this work, all known algebraic traitor tracing schemes relied on variants of the Discrete Logarithm Problem: For instance, the earlier constructions (including [31, 10, 28, 32]) rely on the assumed hardness of the Decisional Diffie Hellman problem (DDH), whereas others (including [17, 13, 14, 1, 20]) 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).

The private key in combinatorial schemes is combined by many sub-keys (ℓ sub-keys) and the traitors can only contribute some sub-keys to produce effective pirate decoders while preserving their anonymity, as shown in Pirates 2.0 [9]. In contrast, the private key in algebraic schemes is indivisible and a partial leakage of the key does not seem to help the pirate to produce pirate decoders. We will concentrate on the algebraic approach in this paper.

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 an outsider of the system. The strongest notion for this requirement 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 [43], a commitment protocol in [44], an oblivious polynomial evaluation in [53, 30, 27].

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, which was put forth in [17], 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 [45, 14]. The scheme [45] is generic, based on IPP-codes and is quite impractical. The Boneh-Waters scheme [14] achieves resistance against unbounded coalitions, but has a large ciphertext size of $O(\sqrt(N))$ group elements. So far, no efficient algebraic scheme in the bounded collusion model enjoys public traceability. In this paper, we achieve public traceability without downgrading the efficiency of the proposed sheme.

Lattice Based Schemes. Since the pioneering work of Ajtai [5], 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 decisional Gap Shortest Vector Problem with polynomial gap (see the surveys [36, 48]). 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 $\leq \mathcal{P}oly(n)$. One can obtain lattice-based traitor tracing schemes by simply using lattice-based encryption within the combinatorial constructions. As discussed above, the efficiency of the resulting schemes is limited.

Our Contributions. We describe the first algebraic construction of a public-key lattice-based traitor tracing scheme. It is semantically secure and achieves public traceability. The security relies on the hardness of the decisional version of the Learning With Errors (LWE) problem, which is known to be at least as hard as standard worst-case lattice problems [47, 40, 15].

For proving the security of the scheme, we introduce the k-LWE problem, which we prove at least as hard as LWE, even for large values of k. 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 [11], 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 (which can be adapted to the SIS framework). 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 [6] (see also Lemma 12). We adapt the Micciancio-Peikert framework from [34] to sampling an integer Gaussian X along with a short basis for the lattice $\ker(X) = \{\boldsymbol{b} : \boldsymbol{b}^t X = \boldsymbol{0}\}$.

Our construction of a traitor tracing scheme from k-LWE can be seen as an additive and noisy variant of the (black-box) Boneh-Franklin traitor tracing scheme [10]. 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 black-box tracing in both the Boneh-Franklin scheme and ours are of high complexity. 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 that our scheme enjoys.

To obtain public traceability, we introduce a new *oblivious sampling in a subspace* technique that allows one to sample a signal from a secret space in an oblivious manner: i) the sampler does not know the secret space although the sampled signal is uniform in a public subspace of it; ii) the adversary, given the secret space, is unable to distinguish between uniform distributions over these two spaces, once some small noise component has been added. This technique can be interpreted as applying the parity-check transformation of Micciancio and Mol [33] to the encryption scheme underlying the Gordon *et al.* group signature scheme [23]. We believe that this technique is interesting in its own and could find other applications.

Open problems. The proposed lattice-based traitor tracing scheme resists Chosen Plaintext Attacks. There exist traitor tracing schemes that resist Chosen Ciphertext Attacks, such as [10, Se. 8], but they rely on more traditional hardness assumptions (such as DDH). It seems quite challenging to devise such an IND-CCA-secure scheme under lattice hardness assumptions. Intuitively, in a traitor tracing scheme the users own parts of a master secret (e.g., each user owns a short vector in a shared lattice, or a discrete log representation with respect to a shared set of group elements), and we attempt to prevent traitors from gaining knowledge of more than their share of the secret information. This requirement seems to be in opposition with the underlying design of all known lattice-based IND-CCA2-secure encryption schemes [39, 42, 16, 2, 3], as the receiver uses the full secret information (a short basis of lattice) to verify the well-formedness of the ciphertext it decrypts. It is an interesting open problem to design an IND-CCA2-secure lattice-based encryption scheme where different independent secret keys could be used for a common public key.

Independently, our scheme naturally raises the question whether the additional properties and features that are enjoyed by existing traitor tracing schemes can also be achieved using lattice hardness assumptions. Our scheme looks as a good starting point for building an ID-based traitor tracing scheme [1], as it seems compatible with the construction from [2]. Another popular functionality is the possibility of revoking malicious users [38] or broadcasting to any subgroup of the users [21]. Our scheme can be directly adapted to broadcast to a small group of users (the tracing signals may be used for sending messages to users who own one of these keys, as if they were suspect keys). This possibility of broadcasting to a small set of the sub-keys can be then combined with the complete subtree framework in [37] to deal with revocation (in which each tree node is associated with a sub-key $\boldsymbol{x}_i$ and the key of each user is a set of sub-keys on the path from the root to the user). However, this leads to a combinatorial and inefficient scheme and can only handle a small set of the revoked users (while maintaining security). An algebraic construction of a lattice-based trace and revoke scheme could be an interesting problem.

Important remark. Due to lack of space, the notations, basic cryptographic definitions and reminders on lattices and lattice problems have been postponed to the appendices. Similarly, most proofs of Section 2 have been postponed to the appendices.

2 New lattice tools

The security of our traitor tracing scheme relies on the hardness of a new problem, which we call (k, S)-LWE$_\alpha$. It is essentially the same as LWE$_\alpha$, except that we give to the distribution distinguisher k vectors sampled from $D_{\Lambda^\perp(A),S}$ (these vectors correspond to the secret keys obtained by the malicious coalition in the traitor tracing scheme). However, this simplified (k, S)-LWE$_\alpha$ becomes easy to solve: take one of the vectors $\boldsymbol{x}$, get a challenge sample $\boldsymbol{y}$ and compute $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \bmod q$; if $\boldsymbol{y}$ was a sample from the LWE distribution $A_{\boldsymbol{s},\alpha}$, then the centred residue of $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \bmod q$ is expected to be of size $\approx \alpha q \sigma_1(S)$, which is $\ll q$ for all parameter settings we will consider; on the other hand, if $\boldsymbol{y}$ was sampled from the uniform distribution, then $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ should be uniform modulo q. We correct the definition of (k, S)-LWE$_\alpha$ by replacing the uniform distribution over $\mathbb{Z}_q^m$ by the uniform distribution over the orthogonal complement of the span of the hint vectors.

In the LWE$_\alpha$ to (k, S)-LWE$_\alpha$ reduction, we use a gadget matrix that has the following properties: it is integral, 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 [6] (see also Lemma 12) to kernel lattices. More precisely, we adapt the Micciancio-Peikert framework from [34] to sampling an integer Gaussian X along with short basis for the lattice $\ker(X) = \{\boldsymbol{b} : \boldsymbol{b}^t X = \boldsymbol{0}\}$.

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

The Micciancio-Peikert construction relies on a *regularity lemma* stating that for $A \hookleftarrow U(\mathbb{Z}_q^{m\times n})$ and $\boldsymbol{r} \hookleftarrow D_{\mathbb{Z}^m,\sigma}$ with sufficiently large standard deviation σ, then the pair $(A, \boldsymbol{r}^t A)$ is statistically close to uniform (over $\mathbb{Z}_q^{m\times(n+1)}$). We use a similar result for an X whose columns are independently sampled from $D_{\mathbb{Z}^m,S}$ for some invertible $S \in \mathbb{R}^{m\times m}$. Such a statement was proven in [4], for $S = \sigma \cdot I_m$ (for a sufficiently large σ). Unfortunately, the result distribution for $\boldsymbol{r}^t \cdot X$ is skewed: its covariance matrix is X^tX. In our applications, we need a covariance matrix for $\boldsymbol{r}^t \cdot X$ that is independent of X. For that, we skew the distribution of $\boldsymbol{r}$ to compensate for the multiplication by X. The following result is an adaptation of the main result from [4] to skewed Gaussian $\boldsymbol{r}$.

Lemma 1. *Fix integer $n \geq 1$, real $\sigma \geq 10\sqrt{\log(100n)/\pi}$, and let m denote an integer such that $m/n \geq 12\log(9m^{1.5}n\sigma)$. Let $X \in \mathbb{Z}^{m\times n}$ be an integral matrix whose entries are sampled from $D_{\mathbb{Z},\sigma}$, let $\boldsymbol{c} \in \mathbb{R}^m$, and let $S \in \mathbb{R}^{m\times m}$ denote an invertible matrix satisfying $\sigma_m(S) \geq 4mn\sqrt{\log(2(m-n)(1+1/\varepsilon))}$ for some $\varepsilon \in (0, 1/2)$. Let $\boldsymbol{r}$ be sampled from $D_{\mathbb{Z}^m,S,\boldsymbol{c}}$, and let $\boldsymbol{z} = X^t \cdot \boldsymbol{r} \in \mathbb{Z}^n$.*

Then, except with probability $2^{-\Omega(m)}$ over the choice of X, the distribution of $\boldsymbol{z}$ (over the choice of $\boldsymbol{r}$) is within statistical distance $\leq 2\varepsilon$ of $D_{\mathbb{Z}^n,SX,X^t\boldsymbol{c}}$.

We now apply Lemma 1 with S invertible chosen such that the covariance matrix $(SX)(SX)^t = \sigma^2 I_m$, thus obtaining an unskewed Gaussian distribution. The scaling σ' is chosen sufficiently large so that the assumptions of Lemmas 16 and 1 hold.

Lemma 2. *We use the notations and assumptions of Lemma 1 and the additional assumption $\sigma \geq \sqrt{m \cdot \log(4m)}$. Let $\sigma' > \sigma \cdot 4m^{1.5}n\sqrt{\log(2(m-n)(1+1/\varepsilon))}$. There exists a ppt algorithm that given n, m, σ, σ' as inputs returns $X \in \mathbb{Z}^{m\times n}, X' \in \mathbb{Z}^{m\times n}, R \in \mathbb{Z}^{m\times m}$ such that: The joint distribution of (X, X') is within $2^{-\Omega(m)}$ statistical distance to $D_{\mathbb{Z},\sigma}^{m\times n} \times D_{\mathbb{Z},\sigma'}^{m\times n}$, we have $(X')^t = [I_n|\mathbf{0}_{n\times(m-n)}] + X^t \cdot R$, and every column of R has norm $\leq O(\sigma'/\sigma)$ with probability $\geq 1 - 2^{-\Omega(m)}$.*

We use the result above to extend the Micciancio-Peikert trapdoor construction to integer kernel lattices. Namely, we run the algorithm from Lemma 2 to get $X_1, X_2 \in \mathbb{Z}^{m\times n}$ and $R \in \mathbb{Z}^{m\times m}$ such that $X_2^t = [I_n|\mathbf{0}_{n\times(m-n)}] + X_1^t \cdot R$ and notice that $U \cdot X = [I_n|\mathbf{0}]^t$, where

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

Note that U is unimodular and has small entries. This leads to the design of our random gadget matrix.

Theorem 1. *Fix integers $n \geq 1$ and $m \geq \Omega(n\log n)$, reals $\sigma \geq \Omega(\sqrt{m\log m})$ and $\sigma' > \sigma \cdot \Omega(m^2 n)$. There exists a ppt algorithm that given n, m, σ, σ' as inputs returns $G \in \mathbb{Z}^{2m\times 2m}$ such that:*

- *The top $n \times 2m$ submatrix of G is within statistical distance $2^{-\Omega(m)}$ of the distribution $D_{\mathbb{Z},\sigma}^{n\times m} \times D_{\mathbb{Z},\sigma'}^{n\times m}$;*
- *G is unimodular and $\|G^{-1}\| \leq O(\sigma' m)$ holds with probability $1 - 2^{-\Omega(m)}$.*

Proof. Set G^{-1} as U^t, where U is the matrix just above. Then the result follows by applying Lemma 2. □

2.2 The k-LWE problem

We define a variant of LWE where the distinguisher is given additional information. It can be seen as the dual of the k-SIS problem from [11]. Let $k \leq m$ and $S \in \mathbb{R}^{m\times m}$ invertible. The (k,S)-LWE$_{\alpha,m}$ problem is as follows: Given $A \hookleftarrow U(\mathbb{Z}_q^{m\times n}), \boldsymbol{u} \hookleftarrow U(\mathbb{Z}_q^{1\times n})$ and $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k \hookleftarrow D_{\Lambda_{-\boldsymbol{u}}^{\perp}(A),S}$, the goal is to distinguish between the distributions (over $\mathbb{T}^{m+1}$)

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

where $A^+ = [\boldsymbol{u}^t | A^t]^t$ and $\boldsymbol{x}_i^+ = (1|\boldsymbol{x}_i^t)^t$ for $i \leq k$. Imposing that the first coordinate is 1 helps ensuring decryption correctness in the traitor tracing scheme. The results of this section carry over to the task of distinguishing between $\frac{1}{q}U(\mathrm{Im}A) + \nu_\alpha^{m+1}$ and $\frac{1}{q}U(\mathrm{Span}_{i\leq k}(\boldsymbol{x}_i)^\perp) + \nu_\alpha^{m+1}$, knowing $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k \hookleftarrow D_{\Lambda^\perp(A),\sigma}$.

Note that if the right hand side distribution had been chosen as $\frac{1}{q}U(\mathbb{Z}_q^{m+1}) + \nu_\alpha^{m+1}$, then it would have been possible to use the given $\boldsymbol{x}_i^+$'s to build a distinguisher: As $\boldsymbol{x}_i^+$ is small, the inner product $\langle \frac{1}{q}\boldsymbol{x}_i^+, \boldsymbol{y}\rangle \bmod 1$ is small when $\boldsymbol{y} \hookleftarrow \frac{1}{q}U\left(\mathrm{Im}A^+\right) + \nu_\alpha^{m+1}$, but it is uniform in $\mathbb{T}$ when $\boldsymbol{y} \hookleftarrow \frac{1}{q}U(\mathbb{Z}_q^{m+1}) + \nu_\alpha^{m+1}$. By using $\frac{1}{q}U(\mathrm{Span}_{i\leq k}(\boldsymbol{x}_i^+)^\perp) + \nu_\alpha^{m+1}$ instead, the inner product $\langle \boldsymbol{x}_i^+, \boldsymbol{y}\rangle$ follows the same distribution in both cases.

The following result shows that this variant of LWE, with S a diagonal matrix, is in fact at least as hard as the original LWE problem. The reduction algorithm 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. Performing this extension without expanding the size of the noise significantly for large k is tricky, and we use the gadget of Theorem 1 for that purpose. The existing approach for this reduction (that we improve below) is the technique used in the SIS to k-SIS reduction from [11]. 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 random matrix A' under the constraint that the hint vectors are in the q-ary lattice $\Lambda^\perp(A')$ of A'. 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 formed by a portion of the hint vectors. Although the entries of G are small, the entries of $\det(G)\cdot G^{-1}$ are almost as large as $\det G$, which is exponential in k and σ. This leads to the 'exponential noise blowup' problem mentioned above. Our novel approach in the reduction below avoids this exponential blowup by sampling the hint vectors using our gadget construction. The latter ensures that the hint vectors are still distributed as small Gaussians but also provides an associated unimodular invertible matrix G with G^{-1} having small entries, thus allowing an efficient reduction, increasing the noise by only a polynomial factor in k, and hence making it useful for both large and small values of k. We believe that this new technique is of independent interest.

Theorem 2. *For any $k \geq 1$, $\ell = \omega(\log n, \log k)$, $w = 2k\ell + \omega(\log n)$, $m \geq 2\cdot(n + (\ell-1)\cdot k + w)$, $q > 2$ prime, let $\delta = \frac{n+(2\ell-1)\cdot k+w}{m+2\ell k}$. Suppose that σ, σ' are such that*

$$\sigma = q^\delta \cdot \omega(\sqrt{(n+\ell\cdot k+w)\cdot\log(m+\ell k)}) \text{ and } \sigma\cdot\Omega(k^3\ell^2) \leq \sigma' \leq (\sigma/(2q^\delta))^{m/k+2\ell},$$

and $q > \sigma'\cdot\Omega(\sqrt{\log(m+\ell k)})$. Then there exists a ppt reduction from $\mathrm{LWE}_{n,m+1,\alpha}$ to (k,S)-$\mathrm{LWE}_{n+(2\ell-1)\cdot k+w, m+2\ell\cdot k, \alpha''}$ for any $\alpha'' = \omega(m(\ell k)^2\sigma\sigma')\cdot\alpha$ and $S = \left[\begin{array}{c|c} \sigma\cdot I_{m+\ell k} & 0 \\ \hline 0 & \sigma'\cdot I_{\ell k}\end{array}\right]$.

The reduction incurs small increases in n and m, and the noise rate α grows by a polynomial factor only. If LWE is exponentially hard for $\alpha = 1/\mathcal{P}oly(n)$, then the reduction preserves this exponential hardness even for large values of k.

Proof. Let $(A^+, \boldsymbol{b})$, denote the given $\mathrm{LWE}_{n,m+1,\alpha}$ input instance, where $A^+ \hookleftarrow U(\mathbb{Z}_q^{m+1})$, and $\boldsymbol{b} \in \mathbb{T}^{m+1}$ comes from either the distribution $\frac{1}{q}U\left(\mathrm{Im}(A^+)\right) + \nu_\alpha^{m+1}$ ('LWE' distribution) or the distribution $\frac{1}{q}U\left(\mathbb{Z}_q^{m+1}\right) + \nu_\alpha^{m+1}$ ('Uniform' distribution). We denote by $\boldsymbol{u} \in \mathbb{Z}_q^{1\times n}$ the first row of A^+ and by $A \in \mathbb{Z}_q^{m\times n}$ the remaining m rows.

The reduction maps $(A^+, \boldsymbol{b})$ to $(A', \boldsymbol{u}', X, \boldsymbol{b}')$ with $A' \in \mathbb{Z}_q^{(m+2\ell k)\times(n+(2\ell-1)\cdot k+w)}$ and $\boldsymbol{u}' \in \mathbb{Z}_q^{n+(2\ell-1)\cdot k+w}$ independent and uniformly random, $X \in \mathbb{Z}^{k\times(m+2\ell k)}$ with its ith row $\boldsymbol{x}_i$ independently sampled from $D_{\Lambda^\perp_{-\boldsymbol{u}'}(A'),S}$ for $i = 1, \ldots, k$, and the distribution of $\boldsymbol{b}' \in \mathbb{T}^{m+1+2\ell k}$ being $\frac{1}{q}U\left(\mathrm{Im}(A')^+\right) + \nu_\alpha^{m+1+2\ell k}$ ('k-LWE' distribution) if $\boldsymbol{b}$ comes from the 'LWE' distribution, and $\frac{1}{q}U\left(\mathrm{Span}_{i\leq k}(\boldsymbol{x}_i^+)^\perp\right)$ ('k-Uniform' distribution) if $\boldsymbol{b}$ comes from the 'Uniform' distribution, where $(A')^+$ is the matrix having $\boldsymbol{u}'$ as its first row and A' as its remaining $m+2\ell k$ rows, and $\boldsymbol{x}_i^+ = (1|\boldsymbol{x}_i^t)^t \in \mathbb{Z}^{m+1+2\ell k}$ for $i = 1, \ldots, k$.

We first give an outline of the reduction to give the main ideas. Then we give the precise reduction.

Step 1: Building the transformation matrix. We build a transformation that sends the LWE left hand side A^+ to the k-LWE left hand side $(A')^+$. Two things are required from this transformation: 1- it outputs some hint vectors $\boldsymbol{x}_i$, distributed as Gaussians; 2- this transform is 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 does not blow up. These two requirements are hard to achieve simultaneously. For this, we use the gadget of the previous subsection.

Let $\overline{X}_2 \in \mathbb{Z}^{2\ell k \times 2\ell k}$ be our gadget returned by the algorithm of Theorem 1 with $n = k$ and $m = \ell k$. The top k rows of $\overline{X}_2$ are distributed as independent Gaussian samples from $D_{\mathbb{Z}^{\ell k},\sigma} \times D_{\mathbb{Z}^{\ell k},\sigma'}$, $\overline{X}_2$ is unimodular, and its inverse is small. Let $\overline{X}_1 \in \mathbb{Z}^{2\ell k \times m}$ be sampled independently from $D_{\mathbb{Z},\sigma}^{2\ell k \times m}$. We define $B \in \mathbb{Z}_q^{2\ell k \times n}$ by:

$$\left(\mathbf{1}|\overline{X}_1|\overline{X}_2\right) \cdot \left(\begin{array}{c} \boldsymbol{u} \\ \hline A \\ \hline B \end{array}\right) = 0 \bmod q.$$

As a consequence, each row of $\left(\overline{X}_1|\overline{X}_2\right)$ belongs to the coset $\Lambda^{\perp}_{-\boldsymbol{u}}(A'')$, where $A'' = [A^t|B^t]^t$ and $(A'')^+ = [\boldsymbol{u}|A^t|B^t]^t = TA^+$, where $T = \begin{bmatrix} I_{m+1} \\ \hline \overline{X}_2^{-1} \cdot (\mathbf{1}|\overline{X}_1) \end{bmatrix} \in \mathbb{Z}^{(m+1+2\ell k)\times(m+1)}$ is the desired 'small' transformation matrix. So it is tempting to define the k-LWE matrix as A'' and give away the top k rows of $\left(\overline{X}_1|\overline{X}_2\right)$ as the k-LWE hint vectors $\boldsymbol{x}_i \in \Lambda^{\perp}_{-\boldsymbol{u}}(A'')$ making up the matrix X. However, this approach doesn't quite work: we have extended A by $2\ell k$ rows, but we give only k hint vectors (we cannot output them all, as the bottom of $\overline{X}_2$ may not be Gaussian). This creates a difficulty for mapping 'Uniform' to 'k-Uniform' in the reduction.

Step 2: Compensating for the extra rows. To solve the above problem, we sample $(2\ell - 1) \cdot k + w$ extra column vectors $C^+ \in \mathbb{Z}_q^{(m+1+2\ell k)\times((2\ell-1)\cdot k+w)}$ that are uniform in the orthogonal mod q of the hint vectors $\boldsymbol{x}_i$, where we choose w sufficiently large to ensure that the columns of $[T|C^+]$ span the full subspace orthogonal to the $\boldsymbol{x}_i$'s mod q. And we finally define $(A')^+ = \left[\begin{array}{c|c} A^+ \\ \hline B \end{array} \middle| C^+\right]$.

Step 3: Compensating for noise skewing. Now, using the above, it will be possible to map 'Uniform' to 'k-Uniform'. 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 it was a "circular" Gaussian (circular means that the covariance matrix is scalar-times-identity) of the form $\alpha^2 \cdot I_{m+1}$, then now the covariant matrix is $\alpha^2 T^t \cdot T$. To compensate for that, we add to $T \cdot \boldsymbol{b}$ a gaussian noise with compensating covariance. Namely, $\Sigma = (\alpha'')^2 \cdot I_{m+1+2\ell k} - \alpha^2 \cdot T^t \cdot T$. We set α'' set sufficiently large to ensure that this symmetric matrix is positive definite.

Reduction description. Overall, on input $(A^+, \boldsymbol{b}) \in \mathbb{Z}_q^{(m+1)\times n} \times \mathbb{T}_{m+1}$, where $A^+ = [\boldsymbol{u}|A^t]^t$, the precise reduction proceeds as follows:

1. Sample gadget $\overline{X}_2 \in \mathbb{Z}^{2\ell k \times 2\ell k}$ as returned by the algorithm of Theorem 1 (with $n = k$ and $m = \ell k$), and sample $\overline{X}_1 \hookleftarrow D_{\mathbb{Z},\sigma}^{2\ell k \times m}$. Define $T = \begin{bmatrix} I_{m+1} \\ \hline \overline{X}_2^{-1} \cdot (\mathbf{1}|\overline{X}_1) \end{bmatrix} \in \mathbb{Z}^{(m+1+2\ell k)\times(m+1)}$. Let X denote the matrix whose ith row $\boldsymbol{x}_i$ is the ith row of $(\overline{X}_1|\overline{X}_2)$, for $i = 1, \ldots, k$.
2. Sample $C^+ \in \mathbb{Z}_q^{(m+1+2\ell k)\times(2\ell-1)\cdot k+w}$ with independent columns uniform orthogonally to X modulo q. Let $\boldsymbol{u}_C \in \mathbb{Z}_q^{(2\ell-1)\cdot k+w}$ denote the top row of C^+, and $C \in \mathbb{Z}_q^{(m+2\ell k)\times(2\ell-1)\cdot k+w}$ denote the remaining $m + 2\ell k$ rows of C^+.
3. Compute $\Sigma = \alpha'' \cdot I_{m+1+2\ell k} - T^t \cdot T$, and $\sqrt{\Sigma}$ such that $\sqrt{\Sigma}^t \cdot \sqrt{\Sigma} = \Sigma$; if Σ is not positive definite, abort.
4. Compute $(A')^+ = (T \cdot A^+|C^+)$ and $\boldsymbol{b'} = T\boldsymbol{b} + \frac{1}{q}C^+ \cdot \boldsymbol{s'} + \sqrt{\Sigma}\boldsymbol{e'}$, $\boldsymbol{s'} \hookleftarrow U(\mathbb{Z}_q^k)$ and $\boldsymbol{e'} \hookleftarrow \nu_1^{m+1+2\ell k}$. Let $\boldsymbol{u'} = [\boldsymbol{u}|\boldsymbol{u}_C] \in \mathbb{Z}_q^{n+(2\ell-1)\cdot k+w}$ denote the top row of $(A')^+$.
5. Return $(A', \boldsymbol{u'}, X, \boldsymbol{b'})$.

The correctness of the reduction now follows from the following three claims.
First, we show, using an extension of Theorem 8, that the pair $(A', \boldsymbol{u'}, X)$ has the correct distribution.

Lemma 3. *The tuple $(A', \boldsymbol{u'}, X)$ is within statistical distance $2^{-\Omega(k\ell)}+2^{-\Omega(n)}$ of the distribution R in which $A' \in \mathbb{Z}_q^{(m+2\ell k)\times(n+(2\ell-1)\cdot k+w)}$ and $\boldsymbol{u'} \in \mathbb{Z}_q^{n+(2\ell-1)\cdot k+w}$ are independent and uniformly random, and $X \in \mathbb{Z}^{k\times(m+2\ell k)}$ has its ith row $\boldsymbol{x}_i$ independently sampled 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' distribution.

Lemma 4. *Assume that $\alpha'' = \omega(m(\ell k)^2\sigma\sigma') \cdot \alpha$. If $\boldsymbol{b}$ is sampled from the 'LWE' distribution $\frac{1}{q}U(\mathrm{Im}A) + \nu_\alpha^{m+1}$ on $\mathbb{T}^{m+1}$, then the distribution of $\boldsymbol{b'}$ is within statistical distance $2^{-\Omega(k\ell)}$ of the 'k-LWE' distribution $\frac{1}{q}U\left(\mathrm{Im}(A')^+\right) + \nu_{\alpha''}^{m+1+2\ell k}$ on $\mathbb{T}^{m+1+2\ell k}$.*

Finally, we consider the 'Uniform' distribution.

Lemma 5. *If $\boldsymbol{b}$ is sampled from the 'Uniform' distribution $\frac{1}{q}U\left(\mathbb{Z}_q^{m+1}\right) + \nu_\alpha^{m+1}$, then the distribution of $\boldsymbol{b'}$ is within statistical distance $2^{2k\ell}q^{-\Omega(w)} + 2^{-\Omega(k\ell)}$ of the 'k-Uniform' distribution $\frac{1}{q}U\left(\mathrm{Span}_{i\leq k}(\boldsymbol{x}_i^+)^\perp\right) + \nu_{\alpha''}^{m+1+2\ell k}$.*

Overall, we have shown a probabilistic polynomial time reduction that maps the 'LWE' distribution to the 'k-LWE' distribution, and the 'Uniform' distribution to the 'k-Uniform' distribution, up to statistical distances $\varepsilon = 2^{-O(n)} + 2^{-\Omega(k\ell)} + 2^{2k\ell}q^{-\Omega(w)}$. By the choice $\ell = \omega(\log n)$ and $w = 2k\ell + \omega(\log n)$, we have that $\varepsilon = n^{-\omega(1)}$ is negligible, so any ppt algorithm with a non-negligible $1/n^{O(1)}$ advantage in solving k-LWE implies a ppt algorithm with a non-negligible advantage in solving LWE. □

3 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.

3.1 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)-LWE$_{m+1,\alpha}$ is hard to solve (see Section 2).

`Setup`. The trusted authority generates a master key pair using the algorithm from Lemma 12. 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 $\mathbb{Z}_q^{1\times n}$. Matrix T will be part of the tracing key tk, whereas the public key is $pk = A^+$, with $A^+ = (\boldsymbol{u}^t | A^t)^t$.

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 7 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 $\boldsymbol{x}_i^+ = (1|\boldsymbol{x}_i^t)^t \in \mathbb{Z}^{m+1}$. By Lemma 8 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 similar to the 1-bit encryption scheme from [22, Se. 7.1], but embedding the plaintext in the least significant bit of the first coordinate of the ciphertext vector. More precisely, the plaintext and ciphertext domains are $\mathcal{P} = \{0, 1\}$ and $\mathcal{C} = \mathbb{Z}_q^{m+1}$ respectively, and:

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

It is a standard observation that this scheme is semantically secure under chosen plaintext attacks (IND-CPA), under the assumption that LWE$_{m+1,\alpha}$ is hard to solve. First, we can scale the LWE challenge samples by q, mapping ν_α to $\nu_{\alpha q}$, and then round the samples to $\mathbb{Z}^{m+1}$, mapping $\nu_{\alpha q}$ to $\lfloor \nu_{\alpha q} \rceil$. As a result, the distributions $U(\mathrm{Im}(A^+)) + \lfloor \nu_{\alpha q} \rceil^{m+1}$ and $U(\mathbb{Z}_q^{m+1})$ are computationally indistinguishable under the LWE hardness assumption, as these transformations can be performed on the LWE samples obliviously to any secret data. Now, as 2 is invertible modulo q, the distributions $U(\mathrm{Im}(A^+))$ and $U(\mathbb{Z}_q^{m+1})$ are preserved by multiplication by 2, and hence the distributions $U(\mathrm{Im}(A^+)) + 2\lfloor \nu_{\alpha q} \rceil^{m+1}$ and $U(\mathbb{Z}_q^{m+1})$ are computationally indistinguishable.

`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 from $\mathbb{Z}_q^{m+1}$ to $\{0, 1\}$:

$$\mathtt{Dec} : \boldsymbol{c} \mapsto (\langle \boldsymbol{x}_i^+, \boldsymbol{c} \rangle \bmod q) \bmod 2.$$

If $\boldsymbol{c}$ is an honestly generated ciphertext of a plaintext $M \in \{0,1\}$, we have $\langle \boldsymbol{x}_i^+, \boldsymbol{c} \rangle = 2\langle \boldsymbol{x}_i^+, \boldsymbol{e} \rangle + M \bmod q$, 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 quantity is itself $\leq 3m\alpha q\sigma'$ for all i, with probability $\geq 1 - N \cdot 2^{-\Omega(n)}$. To ensure that honestly generated ciphertexts decrypt correctly with overwhelming probability, it suffices to set q larger than $4m\alpha q\sigma'$. Note that other parameter constraints will be added to enable tracing.

Theorem 3. *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:*

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

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

3.2 Tracing traitors

We now present a black-box confirmation algorithm `Trace`.[1] 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 $i \in [0, t]$:

$$p_\infty \ = \ \Pr_{\substack{M \hookleftarrow U(\{0,1\}) \\ \boldsymbol{c} \hookleftarrow \mathtt{Enc}(M)}} \left[\mathcal{O}^{\mathcal{D}}(\boldsymbol{c}, M) = 1\right] \text{ and } p_i \ = \ \Pr_{\substack{\boldsymbol{c} \hookleftarrow Tr_i \\ M \hookleftarrow U(\{0,1\})}} \left[\mathcal{O}^{\mathcal{D}}\left(\boldsymbol{c} + \begin{bmatrix} M \\ \boldsymbol{0} \end{bmatrix}, M\right) = 1\right].$$

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

Finally, 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.

3.3 Confirmation and soundness

We start with the confirmation property, whose proof is given in Appendix D.

[1] 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.

Theorem 4. *Assume that decoder $\mathcal{D}$ was built using $\{sk_{i_j}\}_{j\leq k} \subseteq \{sk_i\}_{i\leq t}$. Under the assumption that (t,S)-LWE$_{m+1,\alpha}$ is hard, algorithm* `Trace` *returns "User $\mathcal{U}_i$ is guilty" for some $i \leq t$.*

Proving the soundness property is more involved. We exploit the hardness of (t,S)-LWE and rely on Theorem 8 several times.

Theorem 5. *Assume that decoder $\mathcal{D}$ was built using $\{sk_{i_j}\}_{j\leq k}$. Under the parameter assumptions of Theorem 8 with Theorem 8'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 $i_0 \in \{i_j\}_{j\leq k}$.*

Proof. Assume (by contradiction) that the traitors $\{\mathcal{U}_{i_j}\}_{j\leq k}$ with $k \leq t$ succeed in having `Trace` incriminate an innocent user $\mathcal{U}_{i_0}$ (with $i_0 \notin \{i_j\}_{j\leq k}$). 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.

`Game`$_0$. The game consists of three steps, as follows:

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

We say that $\mathcal{A}$ wins `Game`$_0$ if it finds the value of b with non-negligible advantage.

Algorithm $\mathcal{A}$ can be obtained from algorithm $\mathcal{T}$ by sampling plaintext M uniformly in $\{0,1\}$, and giving $(\boldsymbol{c}+(M|\boldsymbol{0})^t, M)$ as input to $\mathcal{O}^{\mathcal{D}}$, where $\boldsymbol{c}$ is any sample from `Challenge`$_0$.We now introduce two variations of `Game`$_0$, which differ in the Initialize and Challenge steps.

`Game`$_1$. The game consists of three steps, as follows:

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

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

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

`Game`$'_1$ is as `Game`$_1$, except that if $b=0$ in the challenge step, then the samples sent to $\mathcal{A}$ are from the distribution $U\left(\mathrm{Span}_{i\leq i_0}(\boldsymbol{x}_i^+)^\perp\right) + \lfloor \nu_{\alpha q}\rceil^{m+1}$. (The $\boldsymbol{b}_j$'s are sampled from $U(\mathrm{Span}_{i<i_0}(\boldsymbol{x}_i^+)^\perp)$ in both cases.)

Note that $\mathcal{A}$'s inputs in `Game`$_0$, `Game`$_1$ and `Game`$'_1$ are identical (only the distributions of the Challenge steps vary). By the triangle inequality, if $\mathcal{A}$ wins `Game`$_0$ with some non-negligible advantage, then it may be used to win either `Game`$_1$ or `Game`$'_1$ with non-negligible advantage. In our use of $\mathcal{A}$ to solve $(t+1,S)$-LWE, we may guess in which situation we are. We now consider the two situations separately.

First situation: Algorithm $\mathcal{A}$ wins `Game`$_1$ with non-negligible advantage. Then it may be used to solve $(t+1,S)$-LWE. Indeed, assume we have a $(t+1,S)$-LWE input $(A, \boldsymbol{u}, (\boldsymbol{x}_i)_{i\leq t+1})$, and that we aim at distinguishing between the following distributions over $\mathbb{Z}_q^{m+1}$ (with $A^+ = (\boldsymbol{u}^t|A^t)^t$):

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

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

vectors, and Algorithm $\mathcal{A}$ distinguishes between the resulting distributions, and its output is forwarded as output to the initial $(t+1,S)$-LWE instance.

Second situation: Algorithm $\mathcal{A}$ wins $\texttt{Game}'_1$ 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 $A \hookleftarrow U(\mathbb{Z}_q^{m\times n})$, $\boldsymbol{u} \hookleftarrow U(\mathbb{Z}_q^{1\times n})$, $\boldsymbol{b}_j \hookleftarrow U(\mathbb{Z}_q^m)$ 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 8 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 $i < i_0$, $A^+ = (\boldsymbol{u}^t|A^t)^t \hookleftarrow U(\mathbb{Z}_q^{(m+1)\times n})$ conditioned on $(1|(\boldsymbol{x}_i^+)^t)\cdot A^+ = \boldsymbol{0}$ 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 8 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 $A \hookleftarrow U(\mathbb{Z}_q^{m\times n})$, $\boldsymbol{u} \hookleftarrow U(\mathbb{Z}_q^{1\times n})$, $\boldsymbol{b}_j \hookleftarrow U(\mathbb{Z}_q^m)$ and $v_j \hookleftarrow U(\mathbb{Z}_q)$ for $j \le t-i_0+2$, $\boldsymbol{x}_i \hookleftarrow D_{\Lambda^\perp_{-\boldsymbol{u}}(A),\sigma}$ for $i > i_0$ and $\boldsymbol{x}_i \hookleftarrow D_{\Lambda^\perp_{-\boldsymbol{u}'}(A'),\sigma}$ 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 8 twice, but with $(\boldsymbol{x}_i)_{i\le i_0}$: once for swapping the samplings of these $\boldsymbol{x}_i$'s with A^+ and the $\boldsymbol{b}_j^+$'s, and once for swapping the samplings of A^+ and these $\boldsymbol{x}_i$'s. This shows that algorithm $\mathcal{A}$ wins $\texttt{Game}_4$ with non-negligible advantage, where $\texttt{Game}_4$ differs from $\texttt{Game}_3$ only in its first step, as follows.

- $\texttt{Initialize}_4$: Sample $A \hookleftarrow U(\mathbb{Z}_q^{m\times n})$, $\boldsymbol{u} \hookleftarrow U(\mathbb{Z}_q^{1\times n})$, $\boldsymbol{x}_i \hookleftarrow D_{\Lambda^\perp_{-\boldsymbol{u}}(A),\sigma}$ for $i \le t$, and $\boldsymbol{b}_j^+ \hookleftarrow U(\mathrm{Span}_{i\le i_0}(\boldsymbol{x}_i^+)^\perp)$ for $j \le t-i_0+2$.

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

4 Public Traceability

A scheme is publicly traceable if the algorithm $\texttt{Trace}$ can be run using only public data, i.e., the tracing key tk is public. In the scheme of Section 3, the tracing key $tk = (T, (\mathcal{U}_i, \boldsymbol{x}_i)_{i\le N})$ must be kept secret, as it would reveal the secret keys of the users. In this section, we modify tk so that the scheme becomes publicly traceable.

Oblivious sampling in a subspace. Instead of putting the secret key $\boldsymbol{x}_i$ of the ith user in the tracing key, we associate to it a matrix G_i in such a way that, on the one hand, this matrix G_i leaks no significant secret data, and, on the other hand, the matrix G_i can perfectly replace the role of $\boldsymbol{x}_i$ in the tracing signals samplings. These seemingly contradicting goals are achieved by computationally hiding $\boldsymbol{x}_i$ in G_i. More formally, we will show that given the public matrices $(G_j)_{j\le i}$, one can sample a signal that lies in a subspace (up to some small noise) but is also indistinguishable from $U\left(\mathrm{Span}_{j\le i}(\boldsymbol{x}_i^+)^\perp\right) + \lfloor \nu_{\alpha q}\rceil^{m+1}$. We call this the oblivious sampling in a subspace technique. It is the key to transform the tracing procedure of Section 3 into a public tracing procedure.

4.1 The modified scheme

Apart from the tracing key and the tracing signals, the scheme is essentially identical to the one of Section 3. A further difference, required for simulation purposes in the security proof, is that $\sigma' > \sigma$ must be set $\widetilde{\Omega}(\sqrt{mn}+\alpha q)$.

The tracing key.

- Sample $B \in \mathbb{Z}_q^{m\times n}$ uniformly, conditioned on $B^t \cdot A = 0 \pmod q$.
- For each i, sample $\boldsymbol{s}_i \hookleftarrow U(\mathbb{Z}_q^n)$ and compute $\boldsymbol{z}_i = B \cdot \boldsymbol{s_i} + \boldsymbol{x}_i \in \mathbb{Z}_q^m$.
- Set $tk := (B, (\mathcal{U}_i, \boldsymbol{z}_i)_{i\leq N})$. The tracing key is now made public.

Note that the tracing key contains some kind of LWE encryption (with the corresponding public key B) of the secret keys $\boldsymbol{x}_i$. More precisely, the secret keys are used as the LWE noise in publicly revealed LWE samples. This technique is borrowed from [23]. The tracing procedure is as in Section 3.2 except that the tracing signals are now publicly samplable, as described below.

Public sampling of the tracing signals.

- Compute an arbitrary $\mathbb{Z}_q$-basis H for the kernel of B: we have $H^t \cdot B = 0$, and H is $m \times (m-n)$ with probability $\geq 1 - 2^{-\Omega(n)}$. We note that as $B^t \cdot A = 0 \bmod q$, we have $\mathrm{Im}(A) \subseteq \mathrm{Im}(H)$.
- Define $\boldsymbol{h}_i := -\boldsymbol{z}_i^t \cdot H = -\boldsymbol{x}_i^t \cdot H$.
- Each user $\mathcal{U}_i$ is associated to a matrix $G_i = (\boldsymbol{h}_i^t | H^t)^t \in \mathbb{Z}_q^{(m+1)\times(m-n)}$. We have $(\boldsymbol{x}_i^+)^t \cdot G_i = \boldsymbol{0}$ and $\mathrm{Im}(A^+) \subseteq \mathrm{Im}(G_i)$, with $A^+ = (\boldsymbol{u}^t | A^t)^t$.
- The tracing distributions $Tr_0, \ldots, Tr_t$ over $\mathcal{C} = \mathbb{Z}_q^{m+1}$ are now defined as:

$$Tr_i = U\left(\mathbb{Z}_q^{m+1} \cap \mathrm{Im}(G_1) \cap \ldots \cap \mathrm{Im}(G_i)\right) + \lfloor \nu_{\alpha q} \rceil^{m+1}.$$

(Note that $\cap_{j\leq i}\mathrm{Im}(G_j)$ is contained in the former tracing subspace $\mathrm{Span}(\{\boldsymbol{x}_j^+\}_{j\leq i})^\perp$).

The tracing subspace $\mathrm{Im}(G_1) \cap \ldots \cap \mathrm{Im}(G_i)$ (which contains $\mathrm{Im}(A^+)$) can be publicly computed by standard linear algebra techniques, and hence the scheme supports public traceability. For example, one may first compute the sum E of the orthogonal complements of the $\mathrm{Im}(G_i)$'s (note that we have $E = \mathrm{Im}((\boldsymbol{0}^t | B^t)^t) + \mathrm{Span}(\{\boldsymbol{x}_j^+\}_{j\leq i})^\perp$ and $\dim E = n + i$ holds with ovewhelming probability), and then compute $E^\perp$.

4.2 Security proofs

Theorem 6. *Assume a ppt attacker is given the public key $A^+ = (\boldsymbol{u}^t | A^t)^t$, secret keys $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k \hookleftarrow D_{\Lambda^\perp_{-\boldsymbol{u}}(A),\sigma}$, and the tracing key $tk = ((\mathcal{U}_i, \boldsymbol{z}_i)_{i\leq N})$. Under the (k,S)-LWE$_{\alpha,m}$ and LWE$'_{\alpha,m}$ hardness assumptions, it cannot distinguish a correct ciphertext from a tracing signal, i.e., between the distributions (over $\mathbb{Z}_q^{m+1}$)*

$$U(\mathrm{Im}(A^+)) + \lfloor \nu_{\alpha q} \rceil^{m+1} \quad \textit{and} \quad Tr_k = U\left(\mathbb{Z}_q^{m+1} \cap \mathrm{Im}(G_1) \cap \ldots \cap \mathrm{Im}(G_k)\right) + \lfloor \nu_{\alpha q} \rceil^{m+1}.$$

Proof. We proceed by a sequence of games.

Game$_0$: This is the real game between the challenger and the adversary. We let ε_0 denote the adversary's advantage in distinguishing the distributions $U(\mathrm{Im}(A^+)) + \lfloor \nu_{\alpha q} \rceil^{m+1}$ and $Tr_k = U(\mathbb{Z}_q^{m+1} \cap \mathrm{Im}(G_1) \cap \ldots \cap \mathrm{Im}(G_k)) + \lfloor \nu_{\alpha q} \rceil^{m+1}$. We use the same notation ε_n in any game **Game**$_n$ below. The goal is to show that ε_0 is negligible, under the assumption that (k,S)-LWE$_{\alpha,m}$ is hard.

In **Game**$_0$, the challenger generates the public key $(A,\ \boldsymbol{u})$, the keys $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ and the tracing key $tk := (B, (\mathcal{U}_i, \boldsymbol{z}_i)_{i\leq N})$ as in the description of the scheme:

- The matrix $A \in \mathbb{Z}_q^{m\times n}$ along with the trapdoor $T \in \mathbb{Z}^{m\times m}$ and the vectors $\boldsymbol{u} \in \mathbb{Z}_q^{1\times n}, \boldsymbol{x}_i \in \mathbb{Z}^m$ are generated as in the `Setup` of the scheme described in Section 3.1. We recall that the $\boldsymbol{x}_i$'s are sampled (using T) from $D_{\Lambda^\perp_{-\boldsymbol{u}}(A),S}$.
- The matrix $B \in \mathbb{Z}_q^{m\times n}$ is uniformly chosen, conditioned on $B^t \cdot A = 0$.
- For each $i \leq N$, we set $\boldsymbol{z}_i = B\boldsymbol{s}_i + \boldsymbol{x}_i$ with $\boldsymbol{s}_i \hookleftarrow U(\mathbb{Z}_q^n)$.

Game$_1$: In this second game, instead of using the trapdoor T to generate the short vectors $\boldsymbol{x}_i$, we only use it to generate $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ and then simply sample $\boldsymbol{x}_i \in \mathbb{Z}_q^m$ uniformly, conditioned on $\boldsymbol{x}_i^t \cdot A = -\boldsymbol{u}$. All the other parameters are generated in the same way as in **Game**$_0$.

Lemma 6. *Under the* $\mathrm{LWE}'_{\alpha,m}$ *hardness assumption, the quantity* $|\varepsilon_1 - \varepsilon_0|$ *is negligible.*

Proof. Our aim is to reduce $\mathrm{LWE}'_{\alpha,m+1}$ to distinguishing $\mathbf{Game}_1$ and $\mathbf{Game}_0$. Assume we have the following multiple LWE′ input $(B, \boldsymbol{y}_{k+1}, \ldots, \boldsymbol{y}_N)$ where $B \hookleftarrow U(\mathbb{Z}_q^{m\times n})$, and $\boldsymbol{y}_i = B\boldsymbol{s}_i + \boldsymbol{e}_i$ with $\boldsymbol{s}_i \hookleftarrow U(\mathbb{Z}_q^n)$ and either $\boldsymbol{e}_i \hookleftarrow U(\mathbb{Z}_q^m)$ for all i, or $\boldsymbol{e}_i \hookleftarrow D_{\mathbb{Z}^m,\alpha q}$ for all i. Our goal is to decide whether the $\boldsymbol{e}_i$'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 $A^t \cdot B = 0$ and T is a full-rank basis of $\Lambda^\perp(A)$ satisfying $\|T\| \le O(\sqrt{mn \log q \log m})$. This can be performed using [23, Le. 4].
- Sample $\boldsymbol{u} \hookleftarrow U(\mathbb{Z}_q^{1\times n})$ and sample the keys $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k \hookleftarrow D_{\Lambda^\perp_{-\boldsymbol{u}}(A),S}$ by using the trapdoor T as in the `Setup` of the real scheme (this is why $\sigma' > \sigma$ must be set sufficiently large).
- For each $i \in [k+1, N]$, compute $\boldsymbol{u}_i = \boldsymbol{y}_i^t \cdot A \in \mathbb{Z}_q^{1\times n}$. Since $\boldsymbol{y}_i = B \cdot \boldsymbol{s}_i + \boldsymbol{e}_i$, we have $\boldsymbol{u}_i = \boldsymbol{e}_i^t \cdot A$.
- For each $i \in [k+1, N]$, sample $\boldsymbol{e}'_i \hookleftarrow \boldsymbol{c} - \boldsymbol{y}_i + D_{\Lambda^\perp(A),S_2,-\boldsymbol{c}+\boldsymbol{y}_i}$ where $S_2 = \sqrt{S^2 - \alpha^2 q^2 I_m}$ (these are diagonal matrices), using T. Since $\boldsymbol{y}_i - \boldsymbol{e}_i$ belongs to $\Lambda^\perp(A)$, we can rewrite that $\boldsymbol{e}'_i \hookleftarrow \boldsymbol{c} - \boldsymbol{e}_i + D_{\Lambda^\perp(A),S_2,-\boldsymbol{c}+\boldsymbol{e}_i}$.
- For each $i \in [k+1, N]$, compute $\boldsymbol{z}_i = \boldsymbol{y}_i + \boldsymbol{e}'_i$. We now have $\boldsymbol{z}_i^t \cdot A = \boldsymbol{c}^t \cdot A = \boldsymbol{u}$.
- Define $pk = A^+ = (\boldsymbol{u}^t, A^t)^t$ and $tk = (B, (\mathcal{U}_i, \boldsymbol{z}_i)_{i\le N})$.

We observe that for each $i \in [k+1, N]$, we have $\boldsymbol{z}_i = \boldsymbol{y}_i + \boldsymbol{e}'_i = B \cdot s_i + (\boldsymbol{e}_i + \boldsymbol{e}'_i)$. Therefore:

- When $\boldsymbol{e}_i \hookleftarrow D_{\mathbb{Z}^m,\alpha q}$, the residual distribution of $D_{\Lambda^\perp(A),S_2,-\boldsymbol{c}+\boldsymbol{e}_i}$ is within negligible statistical distance to $D_{\Lambda^\perp(A),S,-\boldsymbol{c}}$; this is provided by Lemma 10, whose assumptions are satisfied, thanks to the second lower bound on $\sigma' > \sigma$ (see above the tracing key description) and to Lemma 13; consequently, the residual distribution of $\boldsymbol{e}_i + \boldsymbol{e}'_i$ is negligibly close to $\boldsymbol{c} + D_{\Lambda^\perp(A),S,-\boldsymbol{c}}$, and hence the distribution of $\boldsymbol{z}_i$ is statistically close to $D_{\Lambda^\perp_{\boldsymbol{u}}(A),S}$. Overall, the public data (pk, tk) follows the same distributions as in $\mathbf{Game}_0$.
- When $\boldsymbol{e}_i \hookleftarrow U(\mathbb{Z}_q^m)$, the residual distribution of $\boldsymbol{z}_i$ is uniform (by adapting the argument above). The public data (pk, tk) follows the same distributions as in $\mathbf{Game}_1$.

This completes the proof of the lemma.

$\mathbf{Game}_2$: In this game, the challenger does not need to sample the trapdoor T but it receives a (k,S)-$\mathrm{LWE}_{\alpha,m}$ instance instead. More precisely, the challenger is given $A \hookleftarrow U(\mathbb{Z}_q^{m\times n})$, $\boldsymbol{u} \hookleftarrow \mathbb{Z}_q^{1\times n}$ and $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k \hookleftarrow D_{\Lambda^\perp_{-\boldsymbol{u}}(A),S}$. Let $\boldsymbol{x}_i^+ = (1|\boldsymbol{x}_i^t)^t$ for $i \le k$. It should distinguish between

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

The challenger then uniformly samples $\boldsymbol{x}_i \in \mathbb{Z}_q^m$, for each $i \in [k+1, N]$, conditioned on $\boldsymbol{x}_i^t \cdot A = \boldsymbol{u}$. From $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, the challenger generates the other parameters as in $\mathbf{Game}_1$.

Now, by definition, we have: $\mathrm{Im}(A^+) \subseteq \mathrm{Im}(G_1) \cap \ldots \cap \mathrm{Im}(G_k) \subseteq (\mathrm{Span}_{i\le k}(\boldsymbol{x}_i^+))^\perp$, which implies that:

$$U(\mathrm{Im}A^+) + U(\mathrm{Im}(G_1) \cap \ldots \cap \mathrm{Im}(G_k)) = U(\mathrm{Im}(G_1) \cap \ldots \cap \mathrm{Im}(G_k)),$$
$$U(\mathrm{Span}_{i\le k}(\boldsymbol{x}_i^+)^\perp) + U(\mathrm{Im}(G_1) \cap \ldots \cap \mathrm{Im}(G_k)) = U(\mathrm{Span}_{i\le k}(\boldsymbol{x}_i^+)^\perp).$$

We note that from $G_1, \ldots, G_k$, the challenger can efficiently sample from $U(\mathrm{Im}(G_1) \cap \ldots \cap \mathrm{Im}(G_k))$ and can add the sampled signal to its (k,S)-LWE input samples.

Overall, the challenger can give to the adversary all the data it requires, as in $\mathbf{Game}_1$, and it can transform a (k,S)-$\mathrm{LWE}_{\alpha,m}$ challenge sample to a challenge for the adversary, exactly as in $\mathbf{Game}_1$. This directly implies that $\varepsilon_2 = \varepsilon_1$ and that ε_2 is negligible under the (k,S)-$\mathrm{LWE}_{\alpha,m}$ hardness assumption. □

Finally, because the tracing key is public, we should prove that the knowledge of this extra data provides no significant advantage for an adversary to break the semantic security of the encryption scheme.

Theorem 7. *The scheme is IND-CPA secure under the* $\mathrm{LWE}'_{\alpha,m}$ *and* $\mathrm{LWE}_{\alpha,m}$ *hardness assumptions.*

Proof. As in the basic scheme, the semantic security is guaranteed by the fact that, given all the public information about the system including the public key and the tracing key, the distribution $U(\mathrm{Im}(A^+)) + \lfloor \nu_{\alpha q} \rceil^{m+1}$ remains computationally indistinguishable from uniform. Interestingly, this is a direct consequence of the proof of Theorem 6 for the particular case of $k = 0$. □

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. S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *Proc. of EUROCRYPT*, volume 6110 of *LNCS*, pages 553–572. Springer, 2010. Full version available from the authors upon request.
3. S. Agrawal, D. Boneh, and X. Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In *Proc. of CRYPTO*, volume 6223 of *LNCS*, pages 98–115. Springer, 2010.
4. S. Agrawal, C. Gentry, S. Halevi, and A. Sahai. Sampling discrete gaussians efficiently and obliviously. Cryptology ePrint Archive, Report 2012/714.
5. M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proc. of STOC*, pages 99–108. ACM, 1996.
6. M. Ajtai. Generating hard instances of the short basis problem. In *Proc. of ICALP*, volume 1644 of *LNCS*, pages 1–9. Springer, 1999.
7. J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. *Theor. Comput. Science*, 48(3):535–553, 2011.
8. 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.
9. O. Billet and D. H. Phan. Traitors collaborating in public: Pirates 2.0. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 189–205. Springer, April 2009.
10. 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`.
11. 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`.
12. D. Boneh and M. Naor. Traitor tracing with constant size ciphertext. In *ACM Conference on Computer and Communications Security*, pages 501–510. ACM, 2008.
13. 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.
14. 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.
15. Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. Classical hardness of learning with errors. To appear in the proceedings of STOC 2013.
16. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *Proc. of EUROCRYPT*, volume 6110 of *LNCS*, pages 523–552. Springer, 2010.
17. 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.
18. B. Chor, A. Fiat, and M. Naor. Tracing traitors. In *Proc. of CRYPTO*, volume 839 of *LNCS*, pages 257–270. Springer, 1994.
19. B. Chor, A. Fiat, M. Naor, and B. Pinkas. Tracing traitors. *IEEE Trans. Inf. Th.*, 46(3):893–910, 2000.
20. 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.
21. A. Fiat and M. Naor. Broadcast encryption. In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 480–491. Springer, August 1994.
22. 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`.
23. 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.
24. A. Kiayias and S. Pehlivanglu. *Encryption For Digital Content*. Springer, 2010.
25. 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.
26. 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.
27. A. Kiayias and M. Yung. Breaking and repairing asymmetric public-key traitor tracing. In *Digital Rights Management Workshop*, pages 32–50, 2002.
28. 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.
29. A. Kiayias and M. Yung. Traitor tracing with constant transmission rate. In *Proc. of EUROCRYPT*, volume 2332 of *LNCS*, pages 450–465. Springer, 2002.
30. 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.
31. K. Kurosawa and Y. Desmedt. Optimum traitor tracing and asymmetric schemes. In *Proc. of EUROCRYPT*, LNCS, pages 145–157. Springer, 1998.
32. 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.
33. D. Micciancio and P. Mol. Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In *Proc. of CRYPTO*, volume 6841 of *LNCS*, pages 465–484. Springer, 2011.

34. 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.
35. D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput*, 37(1):267–302, 2007.
36. 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.
37. D. Naor, M. Naor, and J. Lotspiech. Revocation and tracing schemes for stateless receivers. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 41–62. Springer, August 2001.
38. M. Naor and B. Pinkas. Efficient trace and revoke schemes. In *Proc. of Financial Cryptography*, volume 1962 of *LNCS*, pages 1–20. Springer, 2000.
39. C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *Proc. of STOC*, pages 333–342. ACM, 2009.
40. C. Peikert. An efficient and parallel gaussian sampler for lattices. In *Proc. of CRYPTO*, volume 6223 of *LNCS*, pages 80–97. Springer, 2010.
41. C. Peikert, A. Shelat, and A. Smith. Lower bounds for collusion-secure fingerprinting. In *Proc. of SODA*, pages 472–479, 2003.
42. C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In *Proc. of STOC*, pages 187–196. ACM, 2008.
43. B. Pfitzmann. Trials of traced traitors. In *Information Hiding*, volume 1174 of *LNCS*, pages 49–64. Springer, 1996.
44. B. Pfitzmann and M. Waidner. Asymmetric fingerprinting for larger collusions. In *ACM CCS 97*, pages 151–160. ACM Press, April 1997.
45. 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.
46. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proc. of STOC*, pages 84–93. ACM, 2005.
47. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009.
48. O. Regev. The learning with errors problem, 2010. Invited survey in CCC 2010, available at `http://www.cs.tau.ac.il/~odedr/`.
49. 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.
50. 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.
51. 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.
52. G. Tardos. Optimal probabilistic fingerprint codes. *J. ACM*, 55(2), 2008.
53. 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 Notation

If x is a real, 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 $A \in \mathbb{Z}_q^{m \times n}$, we let $\mathrm{Im}(A)$ denote the set $\{A\boldsymbol{s} : \boldsymbol{s} \in \mathbb{Z}_q^n\} \subseteq \mathbb{Z}_q^m$. For $S \subseteq \mathbb{Z}_q^m$, we let $\mathrm{Span}(S)$ denote the set of all linear combinations of elements of S. We let $S^\perp$ denote the linear subspace $\{\boldsymbol{b} \in \mathbb{Z}_q^m : \forall \boldsymbol{c} \in S, \langle \boldsymbol{b}, \boldsymbol{c} \rangle = 0\}$.

If D_1 and D_2 are distributions over a countable set X, their statistical distance $\frac{1}{2}\sum_{x \in X} |D_1(x) - D_2(x)|$ will be denoted by $\Delta(D_1, D_2)$. 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 $\boldsymbol{c} \in \mathbb{R}^n$. 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.

B Reminders

B.1 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 [10]. 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-but-one transform (see [29]). 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 [26, 25].

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.

B.2 Euclidean lattices

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 $\mathbf{0}$ contains n linearly independent vectors of L. The smoothing parameter of L is defined as $\eta_\varepsilon(L) = \min\{r > 0 : \rho_{1/r}(\widehat{L} \setminus \mathbf{0}) \leq \varepsilon\}$ 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 [35, 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 $\boldsymbol{c} \in \mathbb{R}^n$ and an invertible $S \in \mathbb{R}^{m \times m}$, we define the Gaussian distribution of support L, center $\boldsymbol{c}$ and standard deviation S by $D_{L,S,\boldsymbol{c}}(\boldsymbol{b}) \sim \rho_{S,\boldsymbol{c}}(\boldsymbol{b}) = (S^{-t}(\boldsymbol{b} - \boldsymbol{c}))$ for all $\boldsymbol{b} \in L$. When $S = \sigma \cdot I_m$, we simply write $D_{L,\sigma,\boldsymbol{c}}$. Note that $D_{L,S,\boldsymbol{c}} = S^t \cdot D_{S^{-t}L,1,S^{-t}\boldsymbol{c}}$. Gentry et al. [22] gave an algorithm to sample from $D_{L,S,\boldsymbol{c}}$.

Lemma 7 ([15, Le. 2.3]). *There exists a ppt algorithm that, given a basis $(\boldsymbol{b}_i)_i$ of a 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 distributed from $D_{L,S,\boldsymbol{c}}$.*

We will need the following basic results on lattice Gaussians. They are usually stated for full-rank lattices, but we will make use of lattices that are not full-rank. The proofs can be modified readily to handle this more general setup, by relying on an isometry from $\mathrm{Span}(L)$ to $\mathbb{R}^m$.

Lemma 8 (Adapted from [4, Le. 3]). *For any n-dimensional lattice L, $\boldsymbol{c} \in \mathrm{Span}(L)$ and $S \in \mathbb{R}^{m\times m}$ invertible satisfying $\sigma_n(S) \geq \eta_\varepsilon(L)$ 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 9 (Adapted from [35, Le. 4.4]). *For any lattice L, $\boldsymbol{c} \in \mathrm{Span}(L)$ and $S \in \mathbb{R}^{m\times m}$ invertible satisfying $\sigma_n(S) \geq \eta_\varepsilon(L)$ 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 10 (Special case of [40, Th. 3.1]). *Let $S_1, S_2 \in \mathbb{R}^{m\times m}$ invertible, $\boldsymbol{c}$ be arbitrary, and Λ_1, Λ_2 be full-rank lattices with $1 \geq \eta_\varepsilon(S_1^{-1}\Lambda_1)$ and $1 \geq \eta_\varepsilon(\sqrt{S_1^{-2}+S_2^{-2}}\Lambda_2)$ 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 $S = \sqrt{S_1^2+S_2^2}$.*

The following result says that if a lattice $L \subseteq \mathbb{R}^m$ is 'shrunk' to the lattice $M\cdot L$ by applying an invertible linear transformation matrix $M \in \mathbb{R}^{m\times m}$ with largest singular value satisfying $\sigma_1(M) < 1$, then the smoothing parameter of lattice $M\cdot L$ shrinks by a factor $\geq \sigma_1(M)$.

Lemma 11 (Implicit in Le. 3 [4]). *For any lattice $L \subseteq R^m$, $\varepsilon \in (0, 1)$ and invertible matrix $M \in \mathbb{R}^{m\times m}$, we have $\eta_\varepsilon(M\cdot L) \leq \sigma_1(M)\cdot\eta_\varepsilon(L)$.*

B.3 Random lattices

In this work, we use two families of random lattices. The first one, often called q-ary Ajtai lattices, consists in sampling $A \hookleftarrow U(\mathbb{Z}_q^{m\times n})$ and considering $\Lambda^\perp(A) = \{\boldsymbol{x} \in \mathbb{Z}^m : \boldsymbol{x}^t\cdot A = \boldsymbol{0} \bmod q\}$. Similarly, the *kernel integer lattices* are obtained by taking $X \in \mathbb{Z}^{m\times n}$ with columns independently sampled from $D_{\mathbb{Z}^m,S,\boldsymbol{0}}$ and considering $\ker(X) = \{\boldsymbol{b} \in \mathbb{Z}^m : \boldsymbol{b}^t\cdot X = \boldsymbol{0}\}$.

For $A \in \mathbb{Z}_q^{m\times n}$, the lattice $\Lambda^\perp(A)$ is m-dimensional, and a basis can be computed efficiently given A. It is possible to efficiently sample a close to uniform A along with a short basis of $\Lambda^\perp(A)$ (see [6, 7, 40, 34]).

Lemma 12 (Adapted from [7, Th. 3.1]). *There exists a ppt algorithm that given $n, m, q \geq 2$ as inputs samples two matrices $A \in \mathbb{Z}_q^{m\times n}$ and $T \in \mathbb{Z}^{m\times m}$ such that: the distribution of A is within statistical distance $2^{-\Omega(n)}$ from $U(\mathbb{Z}_q^{m\times n})$; the rows of T form a basis of $\Lambda^\perp(A)$; each row of T has norm $\leq 3mq^{n/m}$.*

For $A \in \mathbb{Z}_q^{m\times n}$, $S \in \mathbb{R}^{m\times m}$ invertible and $\boldsymbol{u} \in \mathbb{Z}_q^{1\times n}$, 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 $\boldsymbol{c}^t\cdot A = \boldsymbol{u} \bmod q$. A sample $\boldsymbol{x}$ from $D_{\Lambda_{\boldsymbol{u}}^\perp(A),S}$ can be obtained using Lemma 7 along with the short basis of $\Lambda^\perp(A)$ provided by Lemma 12. Boneh and Freeman [11] showed how to efficiently obtain the residual distribution of $(A, \boldsymbol{x})$ without relying on Lemma 12.

Theorem 8 (Adapted from [11, 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 $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k \in \mathbb{Z}_q^{1\times n}$ 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 $2^{-\Omega(n)}$.*

$$\mathtt{Exp}_0 : \quad A \hookleftarrow U(\mathbb{Z}_q^{m\times n}); \quad \forall i \leq k : \boldsymbol{x}_i \hookleftarrow D_{\Lambda_{\boldsymbol{u}_i}^\perp(A),S}$$

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

This statement generalizes [11, Th. 4.3] in two 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 [11, Th. 4.3], but is important for the encryption scheme from Section 3.1. Second, the condition on m is less restrictive (the corresponding assumption in [11, 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 the following adaptation of the bound from [22].

Lemma 13 (Adapted from [22, Le. 5.3]). *Let q be prime and m, n integers with $m \geq 2n$ and $\varepsilon > 0$. Then, for all except a fraction $2^{-\Omega(n)}$ of $A \in \mathbb{Z}_q^{m \times n}$, we have $\eta_\varepsilon(\Lambda^{\perp}(A)) \leq 4q^{\frac{n}{m}} \sqrt{\log(2m(1+1/\varepsilon))/\pi}$.*

Third, we allow for a non-spherical Gaussian distribution, which is needed in our generalized Micciancio-Peikert trapdoor gadget used in the security reduction from LWE to k-LWE in Sec. 2.2.

We also use the following result on the probability of the Gaussian vectors $\boldsymbol{x}_i$ from Theorem 8 being linearly independent over $\mathbb{Z}_q$.

Lemma 14 (Adapted from [11, Le. 4.5]). *With the notations and assumptions of Theorem 8, the k vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ sampled in $\mathtt{Exp}_1$ and $\mathtt{Exp}_0$ are linearly independent over $\mathbb{Z}_q$, except with probability $2^{-\Omega(n)}$.*

In our k-LWE security reduction, we also need a bound on the probability that a collection of vectors $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_{m+w}$ sampled from a linear subspace X of dimension m over $\mathbb{Z}_q$, spans the full space X. Here, we assume that the first $d \leq m$ vectors $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_d$ are linearly independent but fixed, while the rest are independent and uniformly random in X.

Lemma 15. *Let $q > 2$ prime, $m \geq d$ and w be positive integers. Let E denote an m-dimensional $\mathbb{Z}_q$-linear space. Let $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_{m+w} \in E$, where $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_d$ are (arbitrary) linearly independent and $\boldsymbol{t}_{d+1}, \ldots, \boldsymbol{t}_{m+w}$ are independently sampled from $U(E)$. Then $\mathrm{Span}_{i \leq m+w}(\boldsymbol{t}_i) = E$, with probability $\geq 1 - 2^{m+w-d}/q^{w+1}$.*

Proof. For $i \in [d+1, m+w]$, 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 $r_i = r_{i-1} + \chi_i$ and $r_d = d$, we have $r_{m+w} = d + \sum_{j=d+1}^{m+w} \chi_j$. Thus the condition that $r_{m+w} < m$ is equivalent to $\sum_{j=d+1}^{m+w} \chi_j < m - d$. Let S denote the set of binary vectors of length $m + w - d$ and weight $< m - d$. Then we have to upper bound the probability that $\boldsymbol{\chi} = (\chi_{d+1}, \ldots, \chi_{m+w}) \in S$. To do so, let $\boldsymbol{\chi'} = (\chi'_{d+1}, \ldots, \chi'_{m+w}) \in \{0,1\}^{m+w-d}$ denote any fixed vector in S. Note that for any $i \in [d+1, m+w]$, we have $\Pr[\chi_i = 0 | \chi_j = \chi'_j \text{ for } j = d+1, \ldots, i-1] = q^{d + \sum_{j=d+1}^{i-1} \chi'_j}/q^m \leq 1/q$ since $\boldsymbol{\chi}' \in S$ implies $d + \sum_{j=d+1}^{i-1} \chi'_j < m$. 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 $< m - d$, we have $z > m + w - d - (m - d) = w$, so $\Pr[\boldsymbol{\chi} = \boldsymbol{\chi}'] \leq 1/q^{w+1}$. Taking a union bound over all $\boldsymbol{\chi}' \in S$, and using $|S| \leq 2^{m+w-d}$ completes the proof. □

We now adapt recent results from [4] on the smoothing parameter of kernel integer lattices.

Lemma 16 (Adapted from [4, Cor. 3 and Le. 9]). *Fix integer $n \geq 1$, real $\sigma \geq 10\sqrt{\log(100n)/\pi}$ and let m denote an integer such that $m/n \geq 12 \log(9m^{1.5} n\sigma)$. Let each entry of $X \in \mathbb{Z}^{m \times n}$ be sampled independently from $D_{\mathbb{Z},\sigma}$. Then, except with probability $2^{-\Omega(m)}$, the following hold:*

1. *The $\mathbb{Z}$-span of the rows of X is $\mathbb{Z}^n$.*
2. *We have $\eta_\varepsilon(\ker(X)) \leq 4mn\sqrt{\log(2(m-n)(1+1/\varepsilon))}$ for any $\varepsilon \in (0, 1/2)$.*

Lemma 17 ([4, Le. 8]). *Fix integer $n \geq 1$, $m \geq 2n$, real $\sigma \geq C \cdot \sqrt{n}$ for some absolute constant C. Let each entry of $X \in \mathbb{Z}^{m \times n}$ be sampled independently from $D_{\mathbb{Z},\sigma}$. Then, except with probability $2^{-\Omega(m)}$, we have $\sigma_n(X) \geq \Omega(\sigma\sqrt{m})$.*

B.4 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 $(\boldsymbol{a}, \frac{1}{q}\langle \boldsymbol{a}, \boldsymbol{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{T}$, where $\mathbb{T}$ is $\mathbb{R}$ with addition modulo 1. The (decision) *Learning With Errors problem* LWE_α, introduced by Regev in [46, 47], 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 [47] showed that for $q \leq \mathcal{P}oly(n)$ prime and α in the interval $(\frac{\sqrt{n}}{2q}, 1)$, this problem is (quantumly) at least as hard to solve as standard worst-case lattice problems in dimension n with approximation factors $\mathcal{P}oly(n)/\alpha$. This hardness proof was partly dequantized in [39, 15], and the requirements that q should be prime and bounded by $\mathcal{P}oly(n)$ were waived. In all the following sections, we assume that $q \leq \mathcal{P}oly(n)$ is prime, as this simplifies many technicalities and does not incur efficiency losses.

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., [35, 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 [23] to be no easier than standard LWE (up to a constant factor increase in α).

C Missing proofs of Section 2

C.1 Proof of Lemma 1

The proof is a generalization of the proof of [4, Th. 2], in which $\boldsymbol{r}$ is sampled from a spherical Gaussian centered on the origin (i.e., $S = \sigma \cdot I_m$ and $\boldsymbol{c} = \boldsymbol{0}$).

Proof. By Lemma 16, except with probability $2^{-\Omega(m)}$ over the choice of X, we have that X is a 'good' matrix in the following sense: the $\mathbb{Z}$-span of the rows of X is $\mathbb{Z}^n$ and hence so is the support for the distribution of $\boldsymbol{z}$, and, moreover, we have $\eta_\varepsilon(\ker(X)) \leq \sigma_m(S)$. From now on, we assume that X is 'good', and we let F_X denote the distribution of $\boldsymbol{z}$ over the choice of $\boldsymbol{r}$.

For each $\boldsymbol{z} \in \mathbb{Z}^n$, we have $F_X(\boldsymbol{z}) = D_{\mathbb{Z}^m,S,\boldsymbol{c}}(X_{\boldsymbol{z}}) \sim \rho(S^{-t}(X_{\boldsymbol{z}} - \boldsymbol{c}))$, where $X_{\boldsymbol{z}} = \{\boldsymbol{r} \in \mathbb{Z}^m : X^t\boldsymbol{r} = \boldsymbol{z}\}$ is the set of preimages of $\boldsymbol{z}$ under X^t. This preimage set is a coset of $\ker(X) = \{\boldsymbol{r} \in \mathbb{Z}^m : X^t\boldsymbol{r} = \boldsymbol{0}\}$, i.e., $X_{\boldsymbol{z}} = \boldsymbol{w} + \ker(X)$, where $\boldsymbol{w}$ is an arbitrary fixed element of $X_{\boldsymbol{z}}$. Note that $\ker(X)$ is contained in the subspace $E = \{\boldsymbol{r} \in \mathbb{R}^m : X^t\boldsymbol{r} = \boldsymbol{0}\}$ and hence $X_{\boldsymbol{z}}$ is contained in $E_{\boldsymbol{z}} = \boldsymbol{w} + E = \{\boldsymbol{r} \in \mathbb{R}^m : X^t \cdot \boldsymbol{r} = \boldsymbol{z}\}$.

We now study the quantity $\rho(S^{-t}(X_{\boldsymbol{z}} - \boldsymbol{c}))$. Since $X_{\boldsymbol{z}} \subseteq E_{\boldsymbol{z}}$, the set $S^{-t} \cdot (X_{\boldsymbol{z}} - \boldsymbol{c})$ is contained in the affine space $H_{\boldsymbol{z},\boldsymbol{c}} = S^{-t} \cdot (E_{\boldsymbol{z}} - \boldsymbol{c}) = S^{-t} \cdot (E + \boldsymbol{w} - \boldsymbol{c})$. Let $\boldsymbol{u}_{\boldsymbol{z},\boldsymbol{c}}$ denote the point on $H_{\boldsymbol{z},\boldsymbol{c}}$ of minimal norm $\|\boldsymbol{u}_{\boldsymbol{z},\boldsymbol{c}}\|$. Since $H_{\boldsymbol{z},\boldsymbol{c}} = \{\boldsymbol{v} \in \mathbb{R}^m : X^t \cdot (S^t\boldsymbol{v} + \boldsymbol{c}) = \boldsymbol{z}\} = \{\boldsymbol{v} \in \mathbb{R}^m : (X')^t\boldsymbol{v} = \boldsymbol{z'}\}$, where $X' = S \cdot X$, and $\boldsymbol{z'} = \boldsymbol{z} - X^t\boldsymbol{c}$, we have that $\boldsymbol{u}_{\boldsymbol{z},\boldsymbol{c}} = Y' \cdot \boldsymbol{z'}$, where $Y' = X'((X')^tX)^{-1}$ is the pseudoinverse of X'. For each point $\boldsymbol{r} \in S^{-t} \cdot (X_{\boldsymbol{z}} - \boldsymbol{c}) \subseteq H_{\boldsymbol{z},\boldsymbol{c}}$, by definition of $\boldsymbol{u}_{\boldsymbol{z},\boldsymbol{c}}$, we have $\boldsymbol{r} = \boldsymbol{u}_{\boldsymbol{z},\boldsymbol{c}} + (\boldsymbol{r} - \boldsymbol{u}_{\boldsymbol{z},\boldsymbol{c}})$, where $\boldsymbol{r} - \boldsymbol{u}_{\boldsymbol{z},\boldsymbol{c}}$ is orthogonal to $\boldsymbol{u}_{\boldsymbol{z},\boldsymbol{c}}$. Therefore, $\|\boldsymbol{r}\|^2 = \|\boldsymbol{u}_{\boldsymbol{z},\boldsymbol{c}}\|^2 + \|\boldsymbol{r} - \boldsymbol{u}_{\boldsymbol{z},\boldsymbol{c}}\|^2$, so $\rho(\boldsymbol{r}) = \rho(\boldsymbol{u}_{\boldsymbol{z},\boldsymbol{c}}) \cdot \rho(\boldsymbol{r} - \boldsymbol{u}_{\boldsymbol{z},\boldsymbol{c}})$. Hence, $\rho(S^{-t}(X_{\boldsymbol{z}} - \boldsymbol{c})) = \rho(\boldsymbol{u}_{\boldsymbol{z},\boldsymbol{c}}) \cdot \rho(S^{-t} \cdot (X_{\boldsymbol{z}} - \boldsymbol{c}) - \boldsymbol{u}_{\boldsymbol{z},\boldsymbol{c}})$.

We first study the term $\rho(S^{-t} \cdot (X_{\boldsymbol{z}} - \boldsymbol{c}) - \boldsymbol{u}_{\boldsymbol{z},\boldsymbol{c}})$. We claim that thanks to the choice $\sigma_m(S) \geq \eta_\varepsilon(\ker(X))$, we have $\rho(S^{-t} \cdot (X_{\boldsymbol{z}} - \boldsymbol{c}) - \boldsymbol{u}_{\boldsymbol{z},\boldsymbol{c}}) \in (\frac{1-\varepsilon}{1+\varepsilon}, 1) \cdot \rho(S^{-t} \cdot \ker(X))$, for every $\boldsymbol{z}$. We have $S^{-t} \cdot (X_{\boldsymbol{z}} - \boldsymbol{c}) - \boldsymbol{u}_{\boldsymbol{z},\boldsymbol{c}} = S^{-t} \cdot (\ker(X) + \boldsymbol{w} - \boldsymbol{c}) - \boldsymbol{u}_{\boldsymbol{z},\boldsymbol{c}}$. Since $\ker(X) \subseteq E$ and $\boldsymbol{u}_{\boldsymbol{z},\boldsymbol{c}} \in S^{-t} \cdot (E + \boldsymbol{w} - \boldsymbol{c})$, it follows that $S^{-t} \cdot (X_{\boldsymbol{z}} - \boldsymbol{c}) - \boldsymbol{u}_{\boldsymbol{z},\boldsymbol{c}} = S^{-t} \cdot \ker(X) + \boldsymbol{c'}$ for some $\boldsymbol{c'} \in \mathrm{Span}(S^{-t} \cdot \ker(X))$. Hence, by Lemma 9:

$$\rho(S^{-t} \cdot (X_{\boldsymbol{z}} - \boldsymbol{c}) - \boldsymbol{u}_{\boldsymbol{z},\boldsymbol{c}}) \in \left(\frac{1-\varepsilon}{1+\varepsilon}, 1\right) \cdot \rho(S^{-t} \cdot \ker(X)),$$

if the condition $1 \geq \eta_\varepsilon(S^{-t} \cdot \ker(X))$ holds. We have, by Lemma 11, that $\eta_\varepsilon(S^{-t} \cdot \ker(X)) \leq \sigma_1(S^{-t}) \cdot \eta_\varepsilon(\ker(X)) = \eta_\varepsilon(\ker(X))/\sigma_m(S)$. Therefore, the desired condition is satisfied, since $\sigma_m(S) \geq \eta_\varepsilon(\ker(X))$.

We now analyze the remaining term $\rho(\boldsymbol{u}_{\boldsymbol{z},\boldsymbol{c}})$. Recalling that $\boldsymbol{u}_{\boldsymbol{z},\boldsymbol{c}} = Y' \cdot \boldsymbol{z}'$, we get:

$$\rho\left(\boldsymbol{u}_{\boldsymbol{z},\boldsymbol{c}}\right) = \rho(Y' \cdot \boldsymbol{z}') = \rho(X' \cdot \boldsymbol{z}') = \rho_{SX,X^t\boldsymbol{c}}(\boldsymbol{z}),$$

using that $(Y')^t Y' = ((X')^t X')^{-1}$, since Y' is the pseudoinverse of $X' = SX$, and $\boldsymbol{z}' = \boldsymbol{z} - X^t\boldsymbol{c}$.

Combining the results above, we conclude that $\rho(S^{-t}(X_{\boldsymbol{z}} - \boldsymbol{c})) \in C \cdot (\frac{1-\varepsilon}{1+\varepsilon}) \cdot \rho_{SX,X^t\boldsymbol{c}}(\boldsymbol{z})$ for all $\boldsymbol{z} \in \mathbb{Z}^n$, where $C = \rho(S^{-t} \cdot \ker(X))$ is a constant independent of $\boldsymbol{z}$. It follows that $F_X(\boldsymbol{z}) = \rho_{S,\boldsymbol{c}}(X_{\boldsymbol{z}})/\rho_{S,\boldsymbol{c}}(\mathbb{Z}^m) \in C' \cdot (\frac{1-\varepsilon}{1+\varepsilon}, 1) \cdot D_{SX,X^t\boldsymbol{c}}(\boldsymbol{z})$, where C' is a normalization constant independent of $\boldsymbol{z}$. A standard calculation now shows that the statistical distance between F_X and $D_{SX,X^t\boldsymbol{c}}(\boldsymbol{z})$ is $\leq 1 - \frac{1-\varepsilon}{1+\varepsilon} \leq 2\varepsilon$. □

C.2 Proof of Lemma 2

Proof. We first sample X from $D_{\mathbb{Z},\sigma}^{m \times n}$. By Lemma 16, its row $\mathbb{Z}$-span is $\mathbb{Z}^n$ with probability $\geq 1 - 2^{-\Omega(m)}$: we now assume that we are in this situation. Then we sample R: we sample its ith column from $D_{\mathbb{Z}^m,S,\boldsymbol{c}_i}$ for some invertible matrix $S \in \mathbb{R}^{m \times m}$ and vectors $\boldsymbol{c}_i \in \mathbb{R}^m$ chosen as described below. Finally, we set $(X')^t = [I_n | \mathbf{0}_{n \times (m-n)}] + X^t \cdot R$. If the assumptions of Lemma 1 are satisfied, we know that, except with probability $2^{-\Omega(m)}$ over X, the ith row of X' is, conditioned on X, within statistical distance $2^{-\Omega(m)}$ of $D_{\mathbb{Z}^n,SX,X^t\boldsymbol{c}_i+\boldsymbol{\delta}_i}$, where $\boldsymbol{\delta}_i \in \mathbb{Z}^n$ is the ith unit vector for $i \leq n$, and the zero vector for $i > n$.

For all i, we set $\boldsymbol{c}_i \in \mathbb{Z}^m$ so that $X^t\boldsymbol{c}_i + \boldsymbol{\delta}_i = \mathbf{0}$ (this is possible, as $X^t \cdot \mathbb{R}^m = \mathbb{R}^n$). Now, we build S using the singular value decomposition $U_X \cdot \mathrm{Diag}((\sigma_i(X))_i) \cdot V_X$ of 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 from its decomposition $S = U_S \cdot \mathrm{Diag}((\sigma_i(S))_i) \cdot V_S$: we set $U_S^t = \left[\begin{smallmatrix} V_X & \mathbf{0} \\ \mathbf{0} & I_{m-n} \end{smallmatrix}\right]$ 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 $\sigma_i(S) = \sigma'/\sigma_i(X)$ for $i \leq n$ and $\sigma_i(S) = \sigma_n(S)$ for $i > n$. This leads to $SX = \sigma' \cdot I_n$.

To check that the assumptions of Lemma 1 are satisfied, note that $\sigma_m(S) = \sigma'/\sigma_1(X)$. Hence the assumption $\sigma_m(S) \geq 4mn\sqrt{\log(2(m-n)(1+1/\varepsilon))}$ is satisfied if $\sigma' \geq \sigma_1(X) \cdot 4mn\sqrt{\log(2(m-n)(1+1/\varepsilon))}$. The latter holds by the choice of σ', using the fact that $\sigma_1(X) \leq \|X\| \leq \sqrt{m} \cdot \sigma$, where the second inequality holds with probability $\geq 1 - 2^{-\Omega(m)}$, by Lemma 8.

Finally, the bound on the norm of the columns of R follows from Lemma 8 and the facts that $\sigma_1(S) = \sigma'/\sigma_n(X)$ and $\sigma_n(X) \geq \Omega(\sigma\sqrt{m})$ except with probability $2^{-\Omega(m)}$, by Lemma 17. □

C.3 Proof of Lemma 3

Proof. Let R denote the desired distribution for $(A', \boldsymbol{u}', X)$ defined above. We first apply Theorem 8 (with the Theorem parameters $m, n, \sigma_1(S), \sigma_m(S)$ having the values $m + 2\ell k, n + (2\ell - 1) \cdot k + w$, σ' and σ, respectively) to show that R is within statistical distance $2^{-\Omega(n+\ell k)}$ of the distribution P' on tuples $(A', \boldsymbol{u}', X)$ defined as follows: $\boldsymbol{u}' \in \mathbb{Z}_q^{n+(2\ell-1)\cdot k+w}$ is chosen uniformly at random, $X \in \mathbb{Z}^{k \times (m+2\ell k)}$ has its ith row $\boldsymbol{x}_i$ independently sampled from $D_{\mathbb{Z}^{m+2\ell k},S}$, and $A' \in \mathbb{Z}_q^{(m+2\ell k) \times (n+(2\ell-1)\cdot k+w)}$ is chosen uniformly from the set of solutions to $\boldsymbol{x}_i^t \cdot A' = -\boldsymbol{u}' \bmod q$. Indeed, the assumptions of the lemma are satisfied by our choice of parameters.

Next, let $A' = \left(\frac{A}{B}\middle| C\right)$, where $A \in \mathbb{Z}_q^{m \times n}$, $B \in \mathbb{Z}_q^{2\ell k \times n}$ and $C \in \mathbb{Z}_q^{(m+2\ell k) \times ((2\ell-1)\cdot k+w)}$. Note that in the distribution P', *all* of A' is chosen uniformly from the set of solutions to $X \cdot A' = \boldsymbol{u}' \bmod q$. We now show that P' is within statistical distance $2^{-\Omega(n+\ell k)}$ to the distribution P'' that is defined as P', except that in P'', 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 $X \cdot A' = \boldsymbol{u}' \bmod q$. Indeed, the distribution of $(C, \boldsymbol{u}', X)$ is the same in P' and P'' by definition. The condition on (A, B) in P' is $X_1 \cdot A + X_2 \cdot B = \boldsymbol{u} \bmod q$, where $X_1 \in \mathbb{Z}^{k \times m}$ and $X_2 \in \mathbb{Z}^{k \times 2\ell k}$ are the left and right submatrices of X, respectively. 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 condition has the same number of solutions for $B \in \mathbb{Z}_q^{2\ell k \times n}$ (namely $q^{(2\ell-1)\cdot kn}$). Hence, conditioned on X_2 having rank k, the distribution of (A, B) is the same in P' and P''. Therefore, the statistical distance between P' and P'' is $2^{-\Omega(n+\ell k)}$ if the probability that X_2 has rank k in P' is $2^{-\Omega(n+\ell k)}$. Indeed the latter holds by Lemma 14 and our choice of parameters.

Finally, let P denote the distribution of $(A', \boldsymbol{u}', X)$ in the reduction. We show below that P and P'' are within statistical distance $2^{-\Omega(\ell k)} + 2^{-\Omega(m)}$, which completes the proof.

First, we consider the distribution of X. By Theorem 1 and the choice of ℓ, σ, σ', we have that in distribution P, the last $2\ell k$ columns of X are within statistical distance $\varepsilon_1 = 2^{-\Omega(\ell k)}$ of $D_{\mathbb{Z},\sigma}^{k\times \ell k} \times D_{\mathbb{Z},\sigma'}^{k\times \ell k}$. Since the first m columns of X are independently distributed as $D_{\mathbb{Z},\sigma}^{k\times m}$ in both P and P'', it follows that the distribution of X in P is within statistical distance $\varepsilon_1 = 2^{-\Omega(\ell k)}$ of its distribution $D_{\mathbb{Z}^{m+2\ell k},S}$ in P''.

Next, we consider the distribution of A' given some fixed $(\boldsymbol{u}', X)$. Observe that the only difference between these conditional distributions in P and P'' is that in P, B is defined as the unique solution to $(\mathbf{1}|\overline{\mathbf{X_1}})\cdot(\boldsymbol{u}|A^t)^t + \overline{X_2}\cdot B = 0 \bmod q$, whereas in P'', B is chosen uniformly among the solutions to $(\mathbf{1}|\mathbf{X_1})\cdot(\boldsymbol{u}|A^t)^t + X_2\cdot B = 0 \bmod q$, 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 $\varepsilon_2 = 2^{-\Omega(\ell k)} + 2^{-\Omega(n)}$, which immediately implies that the statistical distance between P and P'' is at most $\varepsilon_1 + \varepsilon_2 = 2^{-\Omega(\ell k)} + 2^{-\Omega(n)}$, as required.

To see this, let X_1', X_2' denote the bottom $(2\ell-1)\cdot k$ rows of $\overline{X}_1, \overline{X}_2$, respectively. Fix $X_1, X_2, X_2', \boldsymbol{u}, A$, with A such that $\eta_{2^{-\ell k}}(\Lambda^\perp(A)) = O(\sqrt{\ell k \log m})\cdot q^{\frac{n}{m}}$. By Lemma 13, this condition holds with probability $1 - 2^{-\Omega(n)}$ over the uniform choice of A. Let B' denote any solution to $(\mathbf{1}|\mathbf{X_1})\cdot(\boldsymbol{u}|A^t)^t + X_2\cdot B = 0 \bmod q$. Let $p(B')$ denote the probability that $B = B'$ in distribution P, conditioned on $X_1, X_2, X_2', \boldsymbol{u}, A$. We show that $p(B')$ is of the form $(1+\varepsilon_{B'})\cdot C$ for any such B', for $\varepsilon_{B'} = 2^{-\Omega(\ell k)}$ and some normalization constant C independent of B'. From this it follows immediately that, in P, the conditional distribution of B is within distance $2^{-\Omega(\ell k)}$ of the uniform distribution on the set of solutions to $(\mathbf{1}|\mathbf{X_1})\cdot(\boldsymbol{u}|\mathbf{A^t})^{\mathbf{t}} + \mathbf{X_2}\cdot\mathbf{B} = \mathbf{0} \bmod \mathbf{q}$, which is the conditional distribution of B in P'', and our claim follows immediately. Let X_1', X_2' denote the bottom $(2\ell-1)\cdot k$ rows of $\overline{X}_1, \overline{X}_2$, respectively. The probability $p(B')$ is the probability that $X_1'\cdot A + X_2'\cdot B = U \bmod q$, conditioned on $X_1, X_2, X_2', \boldsymbol{u}, A$ where U is a $(2\ell-1)\cdot k \times n$ matrix having $-\boldsymbol{u}$ on each row. Let $\boldsymbol{x}'_{1,i} \in \mathbb{Z}^m$ and $\boldsymbol{x}'_{2,i} \in \mathbb{Z}^{2\ell k}$ denote the ith rows of X_1' and X_2', respectively, for $i = 1, \ldots, (2\ell-1)\cdot k$. Observe that the set of solutions for $\boldsymbol{x}'_{1,i} \in \mathbb{Z}^m$ to $\boldsymbol{x}'_{1,i}\cdot A + \boldsymbol{x}'_{2,i}\cdot B' = -\boldsymbol{u} \bmod q$ 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

$$p(B') = \prod_{i=1}^{(2\ell-1)\cdot k} D_{\mathbb{Z}^m,\sigma}(\Lambda^\perp_{-\boldsymbol{u}-\boldsymbol{x}'_{2,i}\cdot B'}(A)).$$

Now, $D_{\mathbb{Z}^m,\sigma}(\Lambda^\perp_{-\boldsymbol{u}-\boldsymbol{x}'_{2,i}\cdot B'}(A)) = \rho_{\sigma,\boldsymbol{c}}(\Lambda^\perp(A))/\rho_\sigma(\mathbb{Z}^m)$ for some $\boldsymbol{c} \in \mathbb{Z}_q^m$ such that $\boldsymbol{c}^t\cdot A = \boldsymbol{u} + \boldsymbol{x}'_{2,i}\cdot B' \bmod q$. By Lemma 9, using the choice of $\sigma \geq \eta_{2^{-\ell k}}(\Lambda^\perp(A)) = O(\sqrt{\ell k \log m})\cdot q^{\frac{n}{m}}$, we have $\rho_{\sigma,\boldsymbol{c}}(\Lambda^\perp(A)) = (1+\varepsilon'_B)\cdot\rho_\sigma(\Lambda^\perp(A))$ for some $\varepsilon'_B = 2^{-\Omega(\ell k)}$. It follows that $p(B') = (1+\varepsilon_B)\cdot\rho_\sigma(\Lambda^\perp(A))$ for some $\varepsilon_B = O(\ell k\cdot 2^{-\Omega(\ell k)}) = 2^{-\Omega(\ell k)}$, as required. □

C.4 Proof of Lemma 4

Proof. In this case, we have $\boldsymbol{b} = \frac{1}{q}A^+\cdot\boldsymbol{s} + \boldsymbol{e}$ with $\boldsymbol{e}$ sampled from ν_α^{m+1} and $\boldsymbol{s}$ uniform in $\mathbb{Z}_q^n$, so

$$\begin{aligned}
\boldsymbol{b}' &= T\cdot\boldsymbol{b} + \frac{1}{q}C^+\cdot\boldsymbol{s}' + \sqrt{\Sigma}\boldsymbol{e}' \\
&= \frac{1}{q}\cdot\frac{1}{q}TA\cdot\boldsymbol{s} + \frac{1}{q}C^+\cdot\boldsymbol{s}' + T\cdot\boldsymbol{e} + \sqrt{\Sigma}\boldsymbol{e}' \\
&= \frac{1}{q}(A')^+\cdot[\boldsymbol{s}|\boldsymbol{s}'] + T\cdot\boldsymbol{e} + \sqrt{\Sigma}\boldsymbol{e}'.
\end{aligned}$$

Now, since $\boldsymbol{s}$ and $\boldsymbol{s}'$ are uniform and independent, we have $\frac{1}{q}(A')^+\cdot[\boldsymbol{s}|\boldsymbol{s}'] \hookleftarrow U(\mathrm{Im}(A')^+$. Moreover, $T\cdot\boldsymbol{e}$ has a continuous Gaussian distribution with covariance matrix $\alpha^2\cdot TT^t$, while $\sqrt{\Sigma}\boldsymbol{e}'$ is independent and has a continuous Gaussian distribution with covariance matrix $\Sigma = (\alpha'')^2 I_{m+1+2\ell k} - \alpha^2 TT^t$ (we show below that Σ is indeed a valid covariance matrix, i.e., is positive definite, so that $\sqrt{\Sigma}$ exists, except with probability $2^{-\Omega(k\ell)}$). Therefore, $T\cdot\boldsymbol{e} + \sqrt{\Sigma}\boldsymbol{e}'$ is distributed as $\nu_{\alpha''}^{m+1+2\ell k}$, as required.

It remains to show that $\Sigma = (\alpha'')^2 I_{m+1+2\ell k} - \alpha^2 TT^t$ is a positive definite matrix, except with probability $2^{-\Omega(k\ell)}$. By definition, the singular values of Σ are of the form $(\alpha'')^2 - \alpha^2\sigma_i(T)^2$, where the $\sigma_i(T)$ are the

singular values of T. It therefore suffices to show that $(\alpha'')^2 - \alpha^2\sigma_1(T)^2 > 0$, where $\sigma_1(T)$ is the largest singular value of T. We have $\sigma_1(T) = \max_{\|\boldsymbol{u}\|=1} \|T\boldsymbol{u}\| \leq \sqrt{m+1+2\ell k}\|T\|$ by Schwarz's inequality, where $\|T\|$ denotes the maximum row norm of T. The first $m+1$ rows of T have norm 1, while the remaining rows have norm $\leq \sqrt{m+1} \cdot \|\overline{X}_2^{-1}\| \cdot t$, where t denotes the maximum column norm of the matrix $(1|\overline{X}_1)$. Since the columns of $\overline{X}_1$ are sampled from $D_{\mathbb{Z}^{2\ell k},\sigma}$, we have by Lemma 8 that $t \leq \sigma \cdot \sqrt{2\ell k}$, and by Theorem 1 that $\|\overline{X}_2^{-1}\| = O(\sigma'\ell k)$, with both bounds holding except with probability $2^{-\Omega(\ell k)}$. It follows that $\sigma_1(T) = O(m(\ell k)^2\sigma\sigma')$, and hence $\alpha'' = \omega(m(\ell k)^2\sigma\sigma') \cdot \alpha$ suffices to ensure that Σ is positive definite, except with probability $2^{-\Omega(\ell k)}$. □

C.5 Proof of Lemma 5

Proof. In this case, we have $\boldsymbol{b} = \frac{1}{q}\boldsymbol{y} + \boldsymbol{e}$ with $\boldsymbol{e}$ sampled from ν_α^{m+1} and $\boldsymbol{y}$ uniform in $\mathbb{Z}_q^{m+1}$, so $\boldsymbol{b'} = \frac{1}{q}T \cdot \boldsymbol{y} + \frac{1}{q}C^+ \cdot \boldsymbol{s'} + T \cdot \boldsymbol{e} + \sqrt{\Sigma}\boldsymbol{e'} = \frac{1}{q}[T|C^+] \cdot [\boldsymbol{y}|\boldsymbol{s'}] + T \cdot \boldsymbol{e} + \sqrt{\Sigma}\boldsymbol{e'}$. Now, since $\boldsymbol{y}$ and $\boldsymbol{s'}$ are uniform and independent, we have $\frac{1}{q}[T|C]^+ \cdot [\boldsymbol{y}|\boldsymbol{s'}] \hookleftarrow U(\mathrm{Im}[T|C^+])$.

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 $2^{-\Omega(w)}$ over the choice of the $\boldsymbol{x}_i$'s and C^+. Indeed, by Lemma 3 and Lemma 14, the vectors $\boldsymbol{x}_1^+, \ldots, \boldsymbol{x}_k^+$ are linearly independent over $\mathbb{Z}_q$ and hence the subspace $X^\perp$ has dimension $m+1+2(\ell-1)\cdot k$, except with probability $2^{-O(n+\ell k+w)}$. Applying Lemma 15 to the subspace $X^\perp$ and the $(m+1+2\ell k) \times (m+1+(2\ell-1)\cdot k+w)$ matrix $[T|C^+]$, using that the rank of T is $m+1$ and that the columns of C are uniform and independent in $X^\perp$, we have that the rank of $[T|C^+]$ is $m+1+(2\ell-1)\cdot k$, and hence $\mathrm{Im}[T|C^+] = X^\perp$, except with probability $\leq 2^{2\ell k+w}q^{-w}$.

Overall, we have shown that $\frac{1}{q}[T|C^+] \cdot [\boldsymbol{y}|\boldsymbol{s'}]$ is within statistical distance $\leq 2^{-\Omega(\ell k)} + 2^{2\ell k}q^{-\Omega(w)}$ of $\frac{1}{q}U(X^\perp)$. As shown in Lemma 4, we also have that the noise term $T \cdot \boldsymbol{e} + \sqrt{\Sigma}\boldsymbol{e'}$ is within statistical distance $2^{-\Omega(\ell k)}$ of the distribution $\nu_\alpha^{m+1+2\ell k}$, as required. □

D Proof of Theorem 4

Wlog we may assume that the traitors in the coalition know all the secret keys $sk_1, \ldots, sk_t$. The hardness of (t,S)-LWE$_{m+1,\alpha}$ implies that the distributions $\mathtt{Enc}(0)$ 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 3.1).

On the other hand, the acceptance probability p_0 is $\leq \frac{1}{2}$. As $p_t - p_0 > \frac{\varepsilon}{2}$ and $|\widetilde{p}_i - p_i| \leq \frac{\varepsilon}{8}$ for all i, we must have $\widetilde{p}_t - \widetilde{p}_0 > \frac{\varepsilon}{4} \geq \frac{\widetilde{\varepsilon}}{8}$, with probability exponentially close to 1. As a consequence, there must exist $i \leq t$ such that $\widetilde{p}_i - \widetilde{p}_{i-1} > \frac{\widetilde{\varepsilon}}{8t}$, and algorithm `Trace` returns "User $\mathcal{U}_i$ is guilty". □