A Lattice-Based Traitor Tracing Scheme

San Ling[1] and Damien Stehlé[2]

[1] Division of Mathematical Sciences,
School of Physical and Mathematical Sciences,
Nanyang Technological University, Singapore
lingsan@ntu.edu.sg – http://www.ntu.edu.sg/home/lingsan/
[2] CNRS, Laboratoire LIP (U. Lyon, CNRS, ENS Lyon, INRIA, UCBL),
46 Allée d'Italie, 69364 Lyon Cedex 07, France.
damien.stehle@gmail.com – http://perso.ens-lyon.fr/damien.stehle

Abstract. 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. All existing traitor tracing schemes rely either on rather inefficient generic constructions from arbitrary encryption schemes and collusion-secure fingerprinting codes, or on algebraic constructions exploiting the assumed 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 assumed (quantum) worst-case hardness of standard lattice problems. The scheme is public-key, provably resists Chosen Plaintext Attacks and allows for minimal access black-box tracing (i.e., tracing works even if granted a very limited access to the pirate decryption device). It inherits the standard features of lattice-based cryptography, such as provable security under mild computational assumptions, conjectured resistance to quantum computers, and asymptotic efficiency. For proving the security, we introduce a Learning With Errors variant of the k-SIS problem from Boneh and Freeman [PKC'11], which we prove at least as hard as the standard LWE problem. We also describe a variant of our scheme with security based on the assumed hardness of the Ring Learning With Errors problem which achieves quasi-optimal asymptotic performance with respect to the security parameter.

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

1 Introduction

A traitor tracing scheme is a multi-receiver encryption scheme with a special functionality aimed at deterring malicious coalitions from building pirate decryption devices. Suppose some malicious users collude and create an unauthorized decryption box. Then the tracing algorithm can use this device to find at least one of the members 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. [14], much work has been devoted to devising efficient and secure traitor tracing schemes. We refer to [21] for an introduction to this rich topic.

There are two main approaches for devising a traitor tracing encryption scheme. Many constructions are combinatorial in nature (see [14, 51, 15, 48, 43, 9], among others): They typically combine an arbitrary encryption scheme with a collusion-resistant fingerprinting code. The efficiency of these traitor tracing schemes is curbed by the large parameters induced by even the best construction of such codes [53]: To resist coalitions of up to t malicious users among N users, the code length is $O(t^2 \log N)$. Lower bounds with the same dependence with respect to t have been given in [39, 53], leaving little hope of significant improvements.

An alternative approach was initiated by Kurosawa and Desmedt in [26] (whose construction was shown insecure in [52]), and quickly improved by Boneh and Franklin [7]: 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 [26, 7, 25]) rely on the assumed hardness of the Decisional Diffie Hellman problem, whereas others (including [12, 10, 1, 18]) 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 main drawback of the algebraic approach compared to the combinatorial one lies in the limited tracing capacity: Most schemes only achieve confirmation tracing (i.e., the tracing algorithm takes as input a bounded number of identities of suspect users), and if the maximum allowed size of traitor coalitions gets large, then the ciphertext length becomes impractical. The combinatorial and algebraic approaches have been combined in a number of works, allowing for circumventing the tracing inefficiency and achieving good transmission rates [24, 18].

Since the pioneering work of Ajtai [4] about 15 years ago, 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 and the computational Shortest Independent Vectors Problem SIVP with polynomial gap and approximation factors respectively (see the recent surveys [34, 46]). 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 $\leq \mathcal{P}oly(n)$. Recently, faster schemes have been obtained by assuming that SIVP remains hard when restricted to the class of ideal lattices (lattices that correspond to ideals in some polynomial rings): These schemes typically achieve quasi-optimal efficiency with respect to the security parameter [32, 29, 38, 28, 50, 30]. 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 lattice-based traitor tracing encryption scheme. It is public-key (i.e., anyone can broadcast messages to the users), semantically secure, and allows for minimal access black-box tracing, thus permitting the tracing algorithm to work successfully with very little access to the pirate device. The security relies on the hardness of the decisional version of the Learning With Errors (LWE) problem, which is known to be (quantumly) at least as hard as standard worst-case lattice problems [45]. This scheme based on LWE is quite inefficient and serves mainly as a proof of concept. To further improve the efficiency, we describe a variant based on the Ring Learning With Errors (RLWE) problem, which was shown in [30] to be (quantumly) at least as hard as SIVP restricted to ideal lattices. Moving from the LWE to the RLWE hardness assumption provides a double speed-up: Matrix-vector multiplications can be replaced by polynomial multiplications, and the plaintext domain can be significantly enlarged. Overall, this results in a traitor tracing scheme where the key-sizes are $\widetilde{O}(\lambda)$, and encryption and decryption of $\widetilde{\Omega}(\lambda)$ plaintext bits cost $\widetilde{O}(\lambda)$ bit operations, where λ is the security parameter (i.e., all known algorithms against the underlying hardness assumption cost $2^{\widetilde{\Omega}(\lambda)}$). The traitor tracing scheme inherits the traditional advantages of lattice-based cryptography. In particular, the low decryption cost could prove an attractive asset for the distribution of copyrighted contents on mobile devices.

For proving the security of the scheme, we introduce the k-LWE problem, which we prove at least as hard as LWE, for small values of k. Intuitively, k-LWE consists in distinguishing between a random vector $\boldsymbol{t}$ close to a given lattice Λ and a random vector $\boldsymbol{t}$ close to the orthogonal of k given short vectors belonging to the dual Λ^* of that lattice. If given $(\boldsymbol{b}_i^*)_{i\leq k}$ small in Λ^*, computing the inner products $\langle \boldsymbol{b}_i^*, \boldsymbol{t}\rangle$ will not help deciding. The k-LWE problem can be interpreted as a dual of the k-SIS problem introduced by Boneh and Freeman [8], which intuitively consists in finding a short vector in Λ^* that is linearly independent with the k given short vectors of Λ^*. Our construction of a traitor tracing scheme from k-LWE can be seen as an additive and noisy variant of the Boneh-Franklin traitor tracing scheme [7].

Related works. This work extends the existing analogies between the LWE and SIS problems on one hand, and the DL and DDH problems on the other hand. SIS can be seen as an additive variant of DL, or, more accurately of the representation problem (which consists in expressing a given group element as a product of powers of other given group elements). For example, the Ajtai and SWIFFT hash functions [4, 29,

38] can be seen as additive counterparts to [13]. In both setups, the hash function can be transformed into a commitment scheme [36, 20]. Similarly, Lyubashevsky's signature [28] is akin to Schnorr's signature [47]. LWE can itself be interpreted as a noisy additive variant of DDH. For instance, the LWE-based encryption schemes from [45, 19, 30] follow a design similar to that of the ElGamal encryption scheme [17].

Open problems. It would be interesting to further investigate the limits of the SIS/LWE–DL/DDH analogy. A major difference is the presence of noise in the LWE problem. For instance, it seems to create a significant obstacle towards designing an LWE adaptation of the Cramer-Shoup IND-CCA2 encryption scheme [16].

Similarly to the proof of hardness of the (k, σ)-SIS problem from [8], our reduction from LWE to (k, σ)-LWE is very loose: The approximation factor for the corresponding worst-case lattice problems increases by a factor $\geq (\sigma k)^k$. If considering the state-of-the-art lattice algorithms, such an approximation factor increase corresponds to taking the k-th root of the run-time. However, for even quite high values of k, it is not known how to exploit the additional information provided in (k, σ)-LWE and (k, σ)-SIS instances to accelerate known LWE and SIS algorithms. An even more surprising aspect of these reductions is that the quality of the reduction (measured by the smallness of the resulting worst-case approximation factor) decreases with σ, although the lower the values of σ, the harder it is to generate the extra input data. Overall, it seems there is room for significantly improving the hardness results of (k, σ)-LWE and (k, σ)-SIS.

The proposed lattice-based traitor tracing encryption scheme resists Chosen Plaintext Attacks. There exist traitor tracing encryption schemes that resist Chosen Ciphertext Attacks, such as [7, Sect. 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 [37, 40, 11, 2, 3], as the receiver uses the full secret information (a short basis of lattice, or a way to generate it himself) 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.

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. For example, our scheme assumes that the tracing authority is trusted, as it could otherwise incriminate an innocent user. Several approaches have been proposed to tackle this issue [41, 42, 25, 12] and it would be interesting to assess whether they can be adapted to our setting. Independently, 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 [35]

2 Reminders

Notation. All vectors will be denoted in bold. By default, vectors will be column vectors. If A is an $m \times n$ matrix over a ring R, then we let $\mathrm{Im}(A)$ denote the set $\{A\boldsymbol{s} : \boldsymbol{s} \in R^n\} \subseteq R^m$. For $S \subseteq R^m$, we let $\mathrm{Span}(S)$ denote the set of all linear combinations of elements of S. We let $\langle \cdot, \cdot \rangle$ denote the canonical inner product over R^m. If R is a field and S is a linear subspace of R^m, then we let $S^\perp$ denote the linear subspace $\{\boldsymbol{b} \in R^m : \forall \boldsymbol{c} \in S, \langle \boldsymbol{b}, \boldsymbol{c} \rangle = 0\}$. For an integer q, we let $\mathbb{Z}_q$ denote the set of integers modulo q.

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 $\rho_{\sigma,\boldsymbol{c}}(\boldsymbol{b}) = \exp(-\pi \|\boldsymbol{b} - \boldsymbol{c}\|/\sigma^2)$ for any $\sigma > 0$ and $\boldsymbol{c} \in \mathbb{R}^n$. We let ν_α denote the one-dimensional Gaussian distribution with standard deviation α.

2.1 Public Key Traitor Tracing Encryption

A public-key traitor tracing scheme consists of four probabilistic algorithms `Setup`, `Encrypt`, `Decrypt`, `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 `Encrypt` 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 `Decrypt` 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.

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

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

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 is run by the trusted authority. 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 [7]. 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 `Encrypt` given pk and M as inputs, then the decryption device decrypts correctly with probability significantly more than $1/|\mathcal{P}|$:

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

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 [24]). 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 [23, 22].

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.

2.2 Euclidean lattices

We will only consider full rank integer lattices, i.e., sets 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 $\mathbb{Z}^n$. In this situation, the $\boldsymbol{b}_i$'s are said to form a basis of the lattice. The n-th minimum $\lambda_n(L)$ of an n-dimensional lattice L is defined as the smallest radius r such that the n-dimensional closed hyperball centered in $\mathbf{0}$ contains n linearly independent vectors of L. We will extensively use the following family of lattices. For $A \in \mathbb{Z}_q^{m\times n}$, we let:

$$\Lambda^{\perp}(A) = \{\boldsymbol{x} \in \mathbb{Z}^m : \boldsymbol{x}^t \cdot A = \mathbf{0} \bmod q\}.$$

This is an m-dimensional lattice, and a basis can be computed efficiently given A. Ajtai, Alwen and Peikert [5, 6] showed how to sample a uniform $A \in \mathbb{Z}_q^{m\times n}$ together with a short basis of $\Lambda^{\perp}(A)$.

Theorem 1 (Adapted from [6, Th. 3.1]). *There exists a polynomial-time algorithm that given $n, m, q \geq 2$ as inputs generates 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 lattice $L \subseteq \mathbb{Z}^n$, a vector $\boldsymbol{c} \in \mathbb{Z}^n$ and a real $\sigma > 0$, we define the Gaussian distribution of support L, center $\boldsymbol{c}$ and standard deviation σ by $D_{L,\sigma,\boldsymbol{c}}(\boldsymbol{b}) = \rho_{\sigma,\boldsymbol{c}}(\boldsymbol{b})/\rho_{\sigma,\boldsymbol{c}}(L)$. Gentry et al. [19] exhibited an algorithm to sample from $D_{L,\sigma,\boldsymbol{c}}$.

Theorem 2 ([19, Th. 4.1]). *There exists a probabilistic polynomial-time algorithm that, given a basis $(\boldsymbol{b}_i)_i$ of an n-dimensional lattice L, $\sigma \geq \sqrt{n} \cdot \max_i \|\boldsymbol{b}_i\|$, and $\boldsymbol{c} \in \mathbb{Z}^n$ as inputs, returns a vector $\boldsymbol{b} \in L$ with distribution within statistical distance $2^{-\Omega(n)}$ to $D_{L,\sigma,\boldsymbol{c}}$.*

For $A \in \mathbb{Z}_q^{m \times n}$, $\sigma > 0$ and $\boldsymbol{u} \in \mathbb{Z}_q^{1 \times n}$, we define the distribution $D_{\Lambda_{\boldsymbol{u}}^\perp(A),\sigma}$ as $\boldsymbol{c} + D_{\Lambda^\perp(A),\sigma,-\boldsymbol{c}}$, 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 $D_{\Lambda_{\boldsymbol{u}}^\perp(A),\sigma}$ can be obtained using Theorem 2 along with the short basis of $\Lambda^\perp(A)$ provided by Theorem 1. Boneh and Freeman [8] showed how to efficiently obtain the residual distribution of $(A, \boldsymbol{x})$ without relying on Theorem 1.

Theorem 3 (Adapted from [8, Th. 4.3]). *Let $n, m, q \geq 2$, $k \geq 0$ and $\sigma > 0$ be such that $n < m - k$, q is prime and $\sigma = \Omega(\sqrt{n} q^{\frac{n}{m-k}})$. 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 from the two following experiments are within statistical distance $2^{-\Omega(n)}$.*

$$\mathtt{Exp}_0 : \quad A \hookleftarrow U(\mathbb{Z}_q^{m \times n}); \quad \boldsymbol{x}_1 \hookleftarrow D_{\Lambda_{\boldsymbol{u}_1}^\perp(A),\sigma}, \ldots, \boldsymbol{x}_k \hookleftarrow D_{\Lambda_{\boldsymbol{u}_k}^\perp(A),\sigma}$$

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

This statement generalizes [8, 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 $\boldsymbol{0}$. Generalizing to arbitrary $\boldsymbol{u}_i$'s does not add any extra complication in the proof of [8, Th. 4.3], but is important for the encryption scheme from Section 4.1. Second, the condition on m is weaker (the corresponding assumption in [8, Th. 4.3] is that $m \geq \max(2n \log q, 2k)$).

We will also need the following basic result on lattice Gaussians.

Lemma 1 ([33, Le. 4.4]). *For any lattice $L \subseteq \mathbb{Z}^n$, $\boldsymbol{c} \in \mathbb{Z}^n$ and $\sigma \geq \sqrt{n} \cdot \lambda_n(L)$, we have $\Pr_{\boldsymbol{b} \hookleftarrow D_{L,\sigma,\boldsymbol{c}}}[\|\boldsymbol{b}\| \geq \sigma\sqrt{n}] \leq 2^{-n+1}$.*

2.3 Learning With Errors

Let $\boldsymbol{s} \in \mathbb{Z}_q^n$ and $\alpha > 0$. We define the distribution $A_{\boldsymbol{s},\alpha}$ as follows: Take $\boldsymbol{a} \hookleftarrow U(\mathbb{Z}_q^n)$ and $e \hookleftarrow \nu_\alpha$, and return $(\boldsymbol{a}, \frac{1}{q}\langle \boldsymbol{a}, \boldsymbol{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{T}$, where $\mathbb{T}$ is $\mathbb{R}$ with addition modulo 1. The (decisional) *Learning With Errors problem* LWE_α, introduced by Regev in [44, 45], 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 [45] 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 with approximation factors $\mathcal{P}oly(n)/\alpha$. In all the following sections, we assume that q is prime.

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., [33, 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.

3 The k-LWE problem

We define a variant of LWE in which the distinguisher is given additional information. It can be seen as the dual of the k-SIS problem from [8]. Let $k \leq m$ and $\sigma > 0$. The (k,σ)-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 $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_k \hookleftarrow D_{\Lambda_{\boldsymbol{u}}^\perp(A),\sigma}$, the goal is to distinguish between the distributions (over $\mathbb{T}^{m+1}$)

$$\frac{1}{q}U\left(\mathrm{Im}\left(\begin{bmatrix}A\\ \boldsymbol{u}\end{bmatrix}\right)\right)+\nu_\alpha^{m+1} \quad \text{and} \quad \frac{1}{q}U\left(\mathrm{Span}(\boldsymbol{x}_1^+,\ldots,\boldsymbol{x}_k^+)^\perp\right)+\nu_\alpha^{m+1},$$

where $\boldsymbol{x}_i^+ \in \mathbb{Z}^{m+1}$ is defined as the vector obtained from $\boldsymbol{x}_i \in \mathbb{Z}^m$, by extending it by one coordinate equal to 1. Alternatively, we could have sampled the $\boldsymbol{x}_i$'s from $\Lambda^\perp(A)$ while requiring their last coordinate to be 1. Imposing that the last coordinate is 1 helps ensuring decryption correctness in the traitor tracing scheme.

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 $\boldsymbol{x}_i^+$'s to build a distinguisher. Indeed, as the vector $\boldsymbol{x}_i^+$ is small, the inner product $\langle \frac{1}{q}\boldsymbol{x}_i^+, \boldsymbol{y}\rangle \bmod 1$ is small when $\boldsymbol{y}$ is sampled from the left hand side distribution, but it is uniform in $\mathbb{T}$ when $\boldsymbol{y}$ is sampled from $\frac{1}{q}U(\mathbb{Z}_q^{m+1})+\nu_\alpha^{m+1}$. 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 seemingly easier variant of LWE is in fact at least as hard as the original LWE problem. In the proof, the hint vectors $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_k$ are built as in the SIS to k-SIS reduction from [8]. The additional difficulty stems from the presence of the noise.

Theorem 4. *Let k,n,m,q be positive integers such that $n < m-k$ and q is prime. Then for any $\sigma,\alpha,\alpha' > 0$ with $\sigma = \Omega(\sqrt{n}q^{\frac{n}{m-k}})$ and $\alpha > (m+1)k!(\sqrt{n}\sigma)^k\alpha'$, there exists a polynomial-time reduction from* LWE$_{\nu_{\alpha'},m-k+1}$ *to* (k,σ)-LWE$_{\nu_\alpha,m}$.

Proof. Let A be sampled uniformly in $\mathbb{Z}_q^{(m-k+1)\times n}$. Our aim is to distinguish between $\frac{1}{q}U(\mathrm{Im}(A))+\nu_{\alpha'}^{m-k+1}$ and $\frac{1}{q}U(\mathbb{Z}_q^{m-k+1})+\nu_{\alpha'}^{m-k+1}$, using an oracle $\mathcal{O}$ that solves the (k,σ)-LWE$_{\nu_\alpha,m}$ problem.

To build the oracle input, we let $\boldsymbol{u}$ be the last row of A and we sample $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_k$ independently from $D_{\mathbb{Z}^m,\sigma}$ (using Theorem 2). We define $X \in \mathbb{Z}^{k\times m}$ as the matrix whose rows are the $\boldsymbol{x}_i^t$'s. By [3, Cor. 32] and the union bound, the rows of X are $\mathbb{Z}_q$-linearly independent, with probability $\geq 1-2k\sigma^{-m+k}$. If this is not the case, the reduction fails (this occurs with exponentially small probability). Wlog, we assume that the last k columns of X are $\mathbb{Z}_q$-linearly independent (and therefore $\mathbb{Q}$-linearly independent). We let $X_1 \in \mathbb{Z}^{k\times(m-k)}$ (resp. $X_2 \in \mathbb{Z}^{k\times k}$) denote the first $m-k$ (resp. last k) columns of X. We then compute $X_2' = -\det(X_2)\cdot X_2^{-1} \in \mathbb{Z}^{k\times k}$ and

$$R = \left[\begin{array}{c|c} I_{m-k} & 0 \\ \hline X_2'\cdot X_1 & X_2'\cdot\begin{pmatrix}1\\ \vdots\\ 1\end{pmatrix}\end{array}\right] \in \mathbb{Z}^{m\times(m-k+1)}.$$

The first inputs to the (k,σ)-LWE$_{\nu_\alpha,m}$ oracle $\mathcal{O}$ are $B,\boldsymbol{u},\boldsymbol{x}_1,\ldots,\boldsymbol{x}_k$, with $B = R\cdot A$. By Theorem 3, the distribution of $(B,\boldsymbol{x}_1,\ldots,\boldsymbol{x}_k)$ is within statistical distance $2^{-\Omega(n)}$ from the distribution obtained by first sampling B uniformly in $\mathbb{Z}_q^{m\times n}$, and then the $\boldsymbol{x}_i$'s from $D_{\Lambda_{\boldsymbol{u}}^\perp(B),\sigma}$.

Now, consider a sample $\boldsymbol{y} \in \mathbb{T}^{m-k+1}$ from either $\frac{1}{q}U(\mathrm{Im}(A))+\nu_{\alpha'}^{m-k+1}$ or $\frac{1}{q}U(\mathbb{Z}_q^{m-k+1})+\nu_{\alpha'}^{m-k+1}$. We are to transform it into a sample $\boldsymbol{z}$ that is to be given to oracle $\mathcal{O}$. For this purpose, we define the following $(m+1)\times(m+1)$ symmetric matrix:

$$\Sigma := \alpha^2 I_{m+1} - S\cdot S^t,$$

where $S \in \mathbb{Z}^{(m+1)\times(m-k+1)}$ is obtained by adjoining the row vector $(0,\ldots,0,1)$ at the end of R. By Lemma 1, each entry of X has magnitude $\leq \sqrt{n}\sigma$, with probability $\geq 1-2^{-\Omega(n)}$. By Cramer's rule, the triangle

inequality and the union bound, each entry of S has magnitude $\leq k!(\sqrt{n}\sigma)^k$ with probability $\geq 1 - 2^{-\Omega(n)}$. As a consequence, each entry of SS^t has magnitude $\leq (m+1)k!^2(\sqrt{n}\sigma)^{2k}$. As $\alpha > (m+1)k!(\sqrt{n}\sigma)^k\alpha'$, we obtain that Σ is diagonally dominant and thus positive definite (with probability exponentially close to 1). We can therefore compute a square-root of Σ, i.e., a matrix $\sqrt{\Sigma}$ such that $\Sigma = \sqrt{\Sigma} \cdot \sqrt{\Sigma}^t$ (for example by using the Cholesky factorization algorithm). Furthermore, the matrix Σ is specifically chosen so that we have equality between the noise distributions

$$S \cdot \nu_{\alpha'}^{m-k+1} + \sqrt{\Sigma} \cdot \nu_1^{m+1} \quad \text{and} \quad \nu_\alpha^{m+1}.$$

Now, we compute $\boldsymbol{z} = S\boldsymbol{y} + \sqrt{\Sigma} \cdot \boldsymbol{e}'$, where $\boldsymbol{e}'$ is sampled from ν_1^{m+1}. If $\boldsymbol{y}$ is sampled from $\frac{1}{q}U(\mathrm{Im}(A)) + \nu_{\alpha'}^{m-k+1}$, then the distribution of $\boldsymbol{z}$ is

$$\frac{1}{q}U\left(\mathrm{Im}\left(\left[\frac{B}{\boldsymbol{u}}\right]\right)\right) + \nu_\alpha^{m+1}.$$

On the other hand, if $\boldsymbol{y}$ is sampled from $\frac{1}{q}U(\mathbb{Z}_q^{m-k+1}) + \nu_{\alpha'}^{m-k+1}$, then the vector $\boldsymbol{z}$ has distribution $\frac{1}{q}U(\mathrm{Span}_{i\leq k}(\boldsymbol{x}_i^+)^\perp) + \nu_\alpha^{m+1}$.

Let us now complete the reduction. When given $(B, \boldsymbol{u}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_k, \boldsymbol{z})$ as input, oracle $\mathcal{O}$ succeeds with some noticeable probability ε, and we forward its reply as output of our LWE algorithm. The arguments above imply that the LWE algorithm succeeds with probability $\geq \varepsilon - 2^{-\Omega(n)}$. □

4 A lattice-based public-key traitor tracing scheme

We describe and analyze the traitor tracing scheme in several steps. First, we give the underlying multi-user public-key encryption scheme. We then explain how to implement minimal access black-box confirmation tracing, and finally prove the soundness and confirmation properties of the tracing algorithm.

4.1 A multi-user encryption scheme

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

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 Theorem 2 using the basis of $\Lambda^\perp(A)$ consisting of the rows of T to obtain a sample $\boldsymbol{x}_i$ from a distribution whose statistical distance from $D_{\Lambda_{\boldsymbol{u}}^\perp(A),\sigma}$ is exponentially small. The standard deviation σ may be chosen as small as $3m^{3/2}q^{n/m}$. The user secret key is $\boldsymbol{x}_i^+ \in \mathbb{Z}^{m+1}$, i.e., vector $\boldsymbol{x}_i$ augmented by one coordinate equal to 1. By Lemma 1, we have $\|\boldsymbol{x}_i\| \leq \sqrt{m}\sigma$ for all $i \leq N$, with probability $\geq 1 - N2^{-m+1}$.

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 [19, Sect. 7.1], but embedding the plaintext in the least significant bit of the last 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 the encryption function is:

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

where α is chosen so that $\alpha q = \Omega(\sqrt{n})$. It is a standard observation that this scheme is semantically secure under chosen plaintext attacks (IND-CPA), under the assumption that $\mathrm{LWE}_{m+1,\alpha}$ is hard to solve. First, the hardness of $\mathrm{LWE}_{m+1,\alpha}$ is preserved when replacing ν_α by the rounding of it to the nearest multiple of $\frac{1}{q}$, because the rounding can be performed on the LWE sample obliviously of any secret data. We can

then multiply the sample by q and we obtain that given A and $\boldsymbol{u}$, the distribution $U\left(\mathrm{Im}\left(\frac{A}{\boldsymbol{u}}\right)\right) + \lfloor \nu_{\alpha q} \rceil$ over $\mathbb{Z}_q^{m+1}$ is computationally indistinguishable from uniform. As 2 is invertible modulo q, this distribution remains computationally indistinguishable from uniform when multiplied by 2. Finally, if $\boldsymbol{s}$ is uniform in $\mathbb{Z}_q^n$, then so is $2\boldsymbol{s}$.

`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 $\{0,1\}$:

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

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 exponentially close to 1 over the randomness of $\boldsymbol{e}$. This quantity is itself $\leq 3m\alpha q\sigma$ for all i, with probability $\geq 1 - N2^{-n+1}$. To ensure that honestly generated ciphertexts decrypt correctly, it suffices to set q larger than $4m\alpha q\sigma$.

Theorem 5. *Let m, n, q, N be integers such that q is prime and $N \leq 2^{o(n)}$. Let $\alpha, \sigma > 0$ such that $\sigma = \Omega(m^{3/2}q^{n/m})$ and $\alpha \in \left(\frac{2\sqrt{n}}{q}, \frac{1}{4m\sigma}\right)$. Then the scheme described above is IND-CPA under the assumption that $\mathrm{LWE}_{m+1,\alpha}$ is hard. Furthermore, the decryption algorithm is correct:*

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

with probability $\geq 1 - 2^{-\Omega(n)}$ over the randomness used in the setup phase and the evaluation of `Enc`.

Note that other parameter constraints will be added to ensure that tracing is possible.

4.2 Tracing traitors

We now present a minimal access black-box confirmation algorithm `Trace`.[3] It is given access to an oracle $\mathcal{O}^{\mathcal{D}}$ that provides minimal 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 the following acceptance probabilities of $\mathcal{O}^{\mathcal{D}}$, for $i \in [0, t]$:

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

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

$$p_\infty \ = \ \Pr_{\substack{M \hookleftarrow U(\{0,1\}) \\ \boldsymbol{c} \hookleftarrow \mathtt{Enc}(M)}} \left[\mathcal{O}^{\mathcal{D}}(\boldsymbol{c}, M) = 1\right].$$

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 $\geq 2^{-\Omega(n)}$ via the Chernoff bound (and this costs $O(\varepsilon^{-2}n)$ calls to $\mathcal{O}^{\mathcal{D}}$).

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

[3] 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. We however keep this formalism, as the RLWE variant from Section 5 handles an exponentially large plaintext domain.

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, `Trace` computes an approximation $\widetilde{p}_i$ of p_i to within an absolute error $\leq 1/(16\widetilde{\varepsilon}t)$, which holds with probability $\geq 1-2^{-n}$. It is obtained via Chernoff's bound by generating $O(\varepsilon^2 n)$ independent samples $\boldsymbol{c}$ from Tr_i.
3. If $\widetilde{p}_i - \widetilde{p}_{i-1} > \frac{\widetilde{\varepsilon}}{8t}$ for some $i \leq t$, then `Trace` returns "$\mathcal{U}_i$ 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.

4.3 Proving the confirmation and soundness properties

We start by proving the confirmation property.

Theorem 6. *Suppose that the setup parameters satisfy $n < m - t$ and $\sigma = \Omega(\sqrt{n} \cdot q^{\frac{n}{m-t}})$. Assume that decoder $\mathcal{D}$ was built using $sk_{i_1}, \ldots, sk_{i_k}$ and that these sk_{i_j}'s all belong to $\{sk_1, \ldots, sk_t\}$. If the problem (t,σ)-$\mathrm{LWE}_{m+1,\alpha}$ is hard to solve, then algorithm* `Trace` *returns "User $\mathcal{U}_i$ is guilty", for some $i \leq t$.*

Proof. Wlog we may assume that the traitors in the coalition know all the secret keys $sk_1, \ldots, sk_t$. The hardness of (t,σ)-$\mathrm{LWE}_{m+1,q,\alpha}$ 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).

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

Proving the soundness property is more involved. We use the hardness of (t,σ)-LWE and Theorem 3 several times.

Theorem 7. *Suppose that the setup parameters satisfy $n < m - 2t$ and $\sigma = \Omega(\sqrt{n+t} \cdot q^{\frac{n+t+1}{m-t-1}})$. Assume that decoder $\mathcal{D}$ was built using $sk_{i_1}, \ldots, sk_{i_k}$. If the problem $(t+1,\sigma)$-$\mathrm{LWE}_{m+1,\alpha}$ is hard to solve, then $i_0 \in \{i_1, \ldots, i_k\}$ for any i_0 such that algorithm* `Trace` *returns "User $\mathcal{U}_{i_0}$ is guilty".*

Proof. Assume (by contradiction) that the traitors $\mathcal{U}_{i_1}, \ldots, \mathcal{U}_{i_k}$ with $k \leq t$ succeed in having `Trace` incriminate an innocent user $\mathcal{U}_{i_0}$ (with $i_0 \notin \{i_1, \ldots, i_t\}$). We are to show that the algorithm $\mathcal{T}$ the traitors use to build the pirate decoder may be exploited for solving $(t+1,\sigma)$-LWE. First, notice that algorithm $\mathcal{T}$ provides an algorithm $\mathcal{A}$ that wins the following game.

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

- Initialize$_0$: Sample $\begin{bmatrix} A \\ \boldsymbol{u} \end{bmatrix} \hookleftarrow U(\mathbb{Z}_q^{(m+1)\times n})$ and $\boldsymbol{x}_i \hookleftarrow D_{\Lambda_{\boldsymbol{u}}^{\perp}(A),\sigma}$ for $i \leq t+1$.
- Input$_0$: Send $A, \boldsymbol{u}$ 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 $\mathtt{Game}_0$ if it finds the value of b with non-negligible advantage.

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

$\texttt{Game}_1$. The game consists of three steps, as follows:

- Initialize$_1$: Sample $\left[\frac{A}{\boldsymbol{u}}\right] \hookleftarrow U(\mathbb{Z}_q^{(m+1)\times n})$, $\boldsymbol{x}_i \hookleftarrow D_{\Lambda_{\boldsymbol{u}}^\perp(A),\sigma}$ for $i \leq t+1$, and $\left[\frac{\boldsymbol{b}_j}{v_j}\right] \hookleftarrow U(\mathrm{Span}_{i<i_0}(\boldsymbol{x}_i^+)^\perp)$ for $j \leq t - i_0 + 2$.
- Input$_1$: Send $A, \boldsymbol{u}$ 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 $U\left(\mathrm{Span}_{i\leq i_0}(\boldsymbol{x}_i^+)^\perp\right) + \lfloor \nu_{\alpha q} \rceil^{m+1}$. If $b = 1$, then send to $\mathcal{A}$ arbitrarily many samples from the distribution:
$$U\left(\mathrm{Im}\left[\frac{A|\boldsymbol{b}_1|\ldots|\boldsymbol{b}_{t-i_0+2}}{\boldsymbol{u}|v_1|\ldots|v_{t-i_0+2}}\right]\right) + \lfloor \nu_{\alpha q} \rceil^{m+1}.$$

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

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

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

First situation: Algorithm $\mathcal{A}$ wins $\texttt{Game}_1'$ with non-negligible advantage. Then it may be used to solve $(t+1,\sigma)$-LWE. Indeed, assume we have a $(t+1,\sigma)$-LWE input $(A, \boldsymbol{u}, (\boldsymbol{x}_i)_{i\leq t+1})$, and that we aim at distinguishing between the following distributions over $\mathbb{Z}_q^{m+1}$:

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

To solve this problem, we sample $\boldsymbol{b}_j$ and v_j for $j \leq t - i_0 + 2$, as in step Initialize$_1$. Then we add a uniform $\mathbb{Z}_q$-linear combination of the $\left[\frac{\boldsymbol{b}_j}{v_j}\right]$'s to the challenge samples. With probability exponentially close to 1 (with respect to n), these vectors are all linearly independent and none of them belongs to $\mathrm{Span}_{i\geq i_0}(\boldsymbol{x}_i^+)$. In that case, the transformation maps the distribution $U\left(\mathrm{Span}_{i\leq t+1}(\boldsymbol{x}_i^+)^\perp\right) + \lfloor \nu_{\alpha q} \rceil^{m+1}$ to the distribution $U\left(\mathrm{Span}_{i<i_0}(\boldsymbol{x}_i^+)^\perp\right) + \lfloor \nu_{\alpha q} \rceil^{m+1}$, and maps the distribution $U\left(\mathrm{Im}\left(\left[\frac{A}{\boldsymbol{u}}\right]\right)\right) + \lfloor \nu_{\alpha q} \rceil^{m+1}$ to the distribution $U\left(\mathrm{Im}\left[\frac{A|\boldsymbol{b}_1|\ldots|\boldsymbol{b}_{t-i_0+2}}{\boldsymbol{u}|v_1|\ldots|v_{t-i_0+2}}\right]\right) + \lfloor \nu_{\alpha q} \rceil^{m+1}$. Algorithm $\mathcal{A}$ thus leads to a $(t+1,\sigma)$-LWE solver.

Second situation: Algorithm $\mathcal{A}$ wins $\texttt{Game}_1$ with non-negligible advantage. Let us define $\texttt{Game}_2$ as being the same as $\texttt{Game}_1$, but with the following new first step:

- Initialize$_2$: Sample $\left[\frac{A}{\boldsymbol{u}}\right] \hookleftarrow U(\mathbb{Z}_q^{(m+1)\times n})$, $\left[\frac{\boldsymbol{b}_j}{v_j}\right] \hookleftarrow U(\mathbb{Z}_q^{m+1})$ for $j \leq t - i_0 + 2$, $\boldsymbol{x}_i \hookleftarrow D_{\Lambda_{\boldsymbol{u}}^\perp(A),\sigma}$ for $i \geq i_0$ and $\boldsymbol{x}_i \hookleftarrow D_{\Lambda_{\boldsymbol{u}'}^\perp(A'),\sigma}$ for $i < i_0$, with
$$A' = [A|\boldsymbol{b}_1|\ldots|\boldsymbol{b}_{t-i_0+2}] \quad \text{and} \quad \boldsymbol{u}' = (\boldsymbol{u}|v_1|\ldots|v_{t-i_0+2}).$$

By using Theorem 3 twice with $(\boldsymbol{x}_i)_{i<i_0}$ (once to swap the sampling of A and that of the $\boldsymbol{x}_i$'s and once to swap the sampling of the $\boldsymbol{x}_i$'s and those of A and the $\boldsymbol{b}_j$'s), we obtain that the residual distributions of $(A, \boldsymbol{u}, (\boldsymbol{b}_j)_j, (v_j)_j, (\boldsymbol{x}_i)_i)$ at the end of Initialize$_1$ and Initialize$_2$ are within exponentially small statistical distance. Therefore, 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 $D_{\Lambda_{\boldsymbol{u}'}^\perp(A'),\sigma}$ (instead of $D_{\Lambda_{\boldsymbol{u}}^\perp(A),\sigma}$).

- Initialize$_3$: Sample $\left[\frac{A}{\boldsymbol{u}}\right] \hookleftarrow U(\mathbb{Z}_q^{(m+1)\times n})$, $\left[\frac{\boldsymbol{b}_j}{v_j}\right] \hookleftarrow U(\mathbb{Z}_q^{m+1})$ for $j \leq t - i_0 + 2$, $\boldsymbol{x}_i \hookleftarrow D_{\Lambda^\perp_{\boldsymbol{u}}(A),\sigma}$ for $i > i_0$ and $\boldsymbol{x}_i \hookleftarrow D_{\Lambda^\perp_{\boldsymbol{u}'}(A'),\sigma}$ for $i \leq i_0$

As $\boldsymbol{x}_{i_0}$ is not given to $\mathcal{A}$ at step Input$_3$, this modification does not alter the winning probability of $\mathcal{A}$, so $\mathcal{A}$ wins $\texttt{Game}_3$ with non-negligible advantage. Now, we again use Theorem 3 twice, but with $(\boldsymbol{x}_i)_{i\leq i_0}$: once for swapping the sampling the $\boldsymbol{x}_i$'s with those of A and the $\boldsymbol{b}_j$'s, and once for swapping the sampling of A and that of the $\boldsymbol{x}_i$'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.

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

The situation we are in now is very similar to that we were in the first situation, when $\mathcal{A}$ was supposed to win $\texttt{Game}'_1$. The arguments used in the first situation carry over here. □

4.4 Example parameters

The following conditions imply that all the assumptions of Theorems 5, 6 and 7 hold:

$$N \leq 2^{o(n)},\ n + 2t < m,\ \sigma = \Omega\left(m^{3/2} q^{\frac{n+t+1}{m-t-1}}\right), \alpha \in \left(\frac{2\sqrt{n}}{q}, \frac{1}{4m\sigma}\right) \text{ and } q \text{ prime.}$$

The security then relies on the assumptions that $\mathrm{LWE}_{m+1,\alpha}$ and $(t+1,\sigma)$-$\mathrm{LWE}_{m+1,\alpha}$ are hard. In order to rely on the worst-case hardness of standard lattice problems with polynomial approximation factors via Theorem 4, one can for example set:

$$N \leq 2^{o(n)},\ t \leq O(1),\ m = cn,\ q = \Omega\left(n^{\frac{3(c-1)}{c-3}}\right),\ \sigma = \Theta\left(n^{3/2} q^{\frac{2}{c-1}}\right) \text{ and } \alpha = \Theta\left(\frac{\sqrt{n}}{q}\right),$$

for any constant $c > 3$. The restriction $t \leq O(1)$ may be an artifact of the looseness of the reduction from LWE to (t,σ)-LWE. If instead the parameters are set to thwart the best known attack against (t,σ)-LWE, the bound t may be set arbitrarily large and m should then increase as $\Omega(n+t)$. With the choice of parameters above, the public key has bit-size $\widetilde{O}(n^2)$ and each user key has bit-size $\widetilde{O}(n)$. Encryption and decryption of 1 bit respectively cost $\widetilde{O}(n^2)$ and $\widetilde{O}(n)$ bit operations.

5 A more efficient scheme based on RLWE

The problem Learning With Errors over Rings introduced in [30] allows for designing more efficient variants of cryptographic protocols based on LWE. The adaptation of our traitor tracing scheme to RLWE is non-trivial and requires some recent results on RLWE. As we will see, the variant of RLWE we use differs from that of [30] in several respects.

The RLWE *problem.* Let $n, q \geq 2$ with n a power of 2. We define the rings $R = \mathbb{Z}[x]/(x^n+1)$ and $R_q = \mathbb{Z}_q[x]/(x^n+1)$. We let $\mathbb{T}_R$ denote $\mathbb{R}[x]/(x^n+1)$ modulo 1. Furthermore, for $\alpha > 0$, we let $\nu_\alpha^{(R)}$ denote the distribution obtained by taking n independent samples from ν_α and interpreting them as the coefficients of a real-valued polynomial modulo x^n+1. For $s \in R_q$ and $\alpha > 0$, we consider the distribution $A_{s,\alpha}^{(R)}$ as the output distribution of the following procedure: Sample $a \hookleftarrow U(R_q)$ and $\boldsymbol{e} \hookleftarrow \nu_\alpha^{(R)}$, and return $(a, \frac{1}{q}a\cdot s + e) \in R_q \times \mathbb{T}_R$. The (decisional) RLWE_α problem consists in secretly sampling $s \hookleftarrow U(R_q)$ and asking for distinguishing between the distributions $A_{s,\alpha}^{(R)}$ and $U(R_q \times \mathbb{T}_R)$ given arbitrarily many samples. In [30], Lyubashevsky et al. showed that under some assumptions (see below), RLWE_α is (quantumly) at least as hard to solve as the Id-SIVP problem with approximation factor $n^{O(1)}/\alpha$, where Id-SIVP is SIVP restricted to lattices that correspond to ideals of R (via the polynomial coefficients to vector coordinates mapping). The result in

fact holds when the distribution $\nu_\alpha^{(R)}$ is replaced by a randomly chosen elliptical Gaussian, with standard deviations within a small polynomial factor to α. The assumptions are that q belongs to $(\frac{\Omega(n^{3/2})}{\alpha}, \mathcal{P}oly(n))$ and that $(x^n+1) \bmod q$ has n distinct linear factors (i.e., $q = 1 \bmod 2n$).

A modified RLWE *problem.* Here we consider RLWE only with a bounded number of samples from $A_{s,\alpha}^{(R)}$. For $m \geq 2$, we define the $\mathrm{RLWE}_{m,\alpha}$ hardness assumption as follows: Given $\boldsymbol{a} \hookleftarrow U(R_q^m)$, the goal is to distinguish between the distributions (over $\mathbb{T}_R^m$)

$$\frac{1}{q}\left(\boldsymbol{a} \cdot U(R_q)\right) + \nu_\alpha^{(R)} \quad \text{and} \quad \frac{1}{q} U\left(R_q^m\right) + \nu_\alpha^{(R)}.$$

Lyubashevsky et al. recently showed in [31] that if the number m of RLWE samples is bounded by a constant, then the result still holds with the noise distribution $\nu_\alpha^{(R)}$. We place ourselves in that situation.

We also modify the choice of the modulus q. Assume that $x^n+1 \bmod q$ factors as $x^n+1 = \prod_{i \leq k} \Phi_i \bmod q$, where the Φ_i's are irreducible. Then, by the Chinese Remainder Theorem, the map reducing an element of R_q modulo each one of the Φ_i's is a ring homomorphism between R_q and the Cartesian product of finite field $\mathbb{F}_{q^{\deg \Phi_i}} \times \ldots \times \mathbb{F}_{q^{\deg \Phi_k}}$. As the proof techniques we use in the case of LWE strongly rely on the fact that $\mathbb{Z}_q$ is a field (ensuring that $\mathbb{Z}_q^m$ is a vector space), in our adaptation to RLWE it would be most convenient to choose a q such that x^n+1 is irreducible modulo q. Unfortunately, such a q does not exist, and the closest we can achieve is to take q so that $x^n+1 \bmod q$ has exactly two irreducible factors, both of degree $n/2$. This is obtained by choosing $q = 3 \bmod 4$ prime. Then $R_q \simeq (\mathbb{F}_q^{n/2})^2$. By a recent result of Langlois and Stehlé [27], RLWE with such a q is still as hard as Id-SIVP with approximation factor $n^{O(1)}/\alpha$.

Main modifications with the LWE*-based approach.* The proofs of Theorems 3, 4 and 7 strongly rely upon $\mathbb{Z}_q^m$ being a vector space over a finite field. It is possible to adapt them to $R_q \simeq (\mathbb{F}_q^{n/2})^2$, which in our situation "behaves" as a field. A typical property that we require is that $k < m$ vectors sampled independently from $U(R_q^m)$ generate a subset of R_q^m of cardinality q^{nk} of R_q^m, with probability $\geq 1 - 2^{-\Omega(n)}$. To show such a property, it suffices to use the Chinese Remainder Theorem and argue that for both copies of $\mathbb{F}_{q^{n/2}}$, the random vectors are linearly independent with probability $\geq 1 - 2^{-\Omega(n)}$.

The other main ingredient in the proof of Theorem 3 is a probabilistic bound on the m-th minimum (or, more precisely, its smoothing parameter) of the lattice $\Lambda^\perp(A)$, for $A \hookleftarrow U(\mathbb{Z}_q^{m \times n})$ (see [8, Le. 4.4]). In our RLWE setting, we instead sample A from $U(R_q^{m \times k})$, for some $k < m$ (the maximum value of k that we use it $t+1$, where t is the a priori bound on the traitor coalition size). The lattice $\Lambda^\perp(A) = \{\boldsymbol{x} \in R^m : \boldsymbol{x}^t \cdot A = \mathbf{0} \bmod q\}$ has dimension mn. It is possible to obtain a bound on its mn-th minimum by adapting the proof of [49, Le. 8] (which only considers the case where $k = 1$).

Lemma 2. *Let n, m, k, q be positive integers with $k \leq m$. We have:*

$$\Pr_{A \hookleftarrow U(R_q^{m \times k})}\left[\lambda_{mn}(\Lambda^\perp(A)) \geq 8mn^{\frac{3}{2}} q^{\frac{k}{m}}\right] \geq 1 - \left(\frac{1}{2\sqrt{n}}\right)^{nk}.$$

A proof of Lemma 2 is given in appendix. Overall, this allows us to obtain the following equivalent to Theorem 3. A proof will be given in the full version.

Theorem 8. *Let $n, d, m, q \geq 2$, $k \geq 0$ and $\sigma > 0$ be such that n is a power of 2, $q = 3 \bmod 4$ is prime, $k + d < m$, q and $\sigma = \Omega(\sqrt{n} q^{\frac{n}{2(m-k)}})$. Let $u_1, \ldots, u_k \in R_q$ be arbitrary. Then the residual distributions of the tuple $(A, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)$ obtained from the two following experiments are within statistical distance $2^{-\Omega(n)}$.*

$$\mathtt{Exp}_0 : \quad A \hookleftarrow U(R_q^{m \times d}); \quad \boldsymbol{x}_1 \hookleftarrow D_{\Lambda_{u_1}^\perp(A),\sigma}, \ldots, \boldsymbol{x}_k \hookleftarrow D_{\Lambda_{u_k}^\perp(A),\sigma}$$

$$\mathtt{Exp}_1 : \boldsymbol{x}_1, \ldots, \boldsymbol{x}_k \hookleftarrow D_{R^m,\sigma};\ A \hookleftarrow U\left(R_q^{m \times d} | \forall i \leq k : \boldsymbol{x}_i^t \cdot A = u_i \bmod q\right).$$

Thanks to the choice of the noise distribution $\nu_\alpha^{(R)}$, the proof of Theorem 4 carries over directly to the RLWE setting, providing a reduction from $\text{RLWE}_{\alpha'}$ to (k,σ)-RLWE_α for $\alpha > k!(\mathcal{P}oly(mn)\sigma)^k\alpha'$.
The RLWE*-based traitor tracing scheme.* The above adaptations of the LWE-based approach allow us to prove the security, soundness and confirmation of the following RLWE-based traitor tracing scheme. In the `Setup` phase, we use the Stehlé et al. [50, Th. 2] adaptation of the Ajtai-Alwen-Peikert algorithm to the ring setting. The public key consists of a matrix $A \in R_q^{m\times 1}$ and a ring element $u \in R_q$. A user secret key is a short vector $\boldsymbol{x} \in R^m$ such that $\boldsymbol{x}^t \cdot A = u \bmod q$, sampled using Theorem 2. The plaintext domain is $\mathcal{P} = \mathbb{Z}_2[x]/(x^n+1)$ and the ciphertext domain is $\mathcal{C} = R_q^{m+1}$. Encryption is as in the LWE case, and decryption of a ciphertext $\boldsymbol{c} \in \mathcal{C}$ with secret key $\boldsymbol{x}$ is performed by computing $(\langle \boldsymbol{x}^+, \boldsymbol{c}\rangle \bmod q) \bmod 2$ where $\boldsymbol{x}^+ \in R^{m+1}$ is $\boldsymbol{x}$ augmented with one coordinate equal to 1. Tracing is as in the LWE setting. This scheme achieves quasi-optimal efficiency: For $t, m = O(1)$, the keys have bit-sizes $\widetilde{O}(n)$ and encryption/decryption of n plaintext bits costs $\widetilde{O}(n)$; and on the other hand the best known attacks against the underlying worst-case hardness assumption costs $2^{-\widetilde{\Omega}(n)}$. Finally, note that constant transmission rate $|\mathcal{C}|/|\mathcal{P}|$ can be achieved for $t = O(1)$, by replacing the modulus 2 by an integer p of bit-length proportional to q (e.g., taking $p = \Theta(\sqrt{q})$).

Acknowledgements. We thank D. Augot, G. Hanrot, F. Laguillaumie, K. T. T. Nguyen, D. H. Phan, G. Quintin, O. Regev, R. Steinfeld and H. Wang for helpful discussions. The authors were partly supported by the LaBaCry MERLION grant, the Australian Research Council Discovery Grant DP110100628, the INRIA invited researcher scheme, and the Singapore National Research Foundation Research Grant NRF-CRP2-2007-03.

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. M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proc. of STOC*, pages 99–108. ACM, 1996.
5. M. Ajtai. Generating hard instances of the short basis problem. In *Proc. of ICALP*, volume 1644 of *LNCS*, pages 1–9. Springer, 1999.
6. J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. *Theor. Comput. Science*, 48(3):535–553, 2011.
7. 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`.
8. 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`.
9. D. Boneh and M. Naor. Traitor tracing with constant size ciphertext. In *ACM Conference on Computer and Communications Security*, pages 501–510. ACM, 2008.
10. D. Boneh, A. Sahai, and B. Waters. Fully collusion resistant traitor tracing with short ciphertexts and private keys. In *Proc. of EUROCRYPT*, volume 4004 of *LNCS*, pages 573–592. Springer, 2006.
11. D. 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.
12. 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.
13. D. Chaum, E. van Heijst, and B. Pfitzmann. Cryptographically strong undeniable signatures, unconditionally secure for the signer. In *Proc. of CRYPTO*, volume 576 of *LNCS*, pages 470–484. Springer, 1991.
14. B. Chor, A. Fiat, and M. Naor. Tracing traitors. In *Proc. of CRYPTO*, volume 839 of *LNCS*, pages 257–270. Springer, 1994.

15. B. Chor, A. Fiat, M. Naor, and B. Pinkas. Tracing traitors. *IEEE Trans. Inf. Th.*, 46(3):893–910, 2000.
16. R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Proc. of CRYPTO*, volume 1462 of *LNCS*, pages 13–25. Springer, 1998.
17. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inf. Th.*, 31(4):469–472, 1985.
18. 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.
19. 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`.
20. A. Kawachi, K. Tanaka, and K. Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *Proc. of ASIACRYPT*, volume 5350 of *LNCS*, pages 372–389. Springer, 2008.
21. A. Kiayias and S. Pehlivanglu. *Encryption For Digital Content*. Springer, 2010.
22. 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.
23. 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.
24. A. Kiayias and M. Yung. Traitor tracing with constant transmission rate. In *Proc. of EUROCRYPT*, volume 2332 of *LNCS*, pages 450–465. Springer, 2002.
25. A. Kiayias and M. Yung. Breaking and repairing asymmetric public-key traitor tracing. In *Proc. of DRM workshop*, volume 2696 of *LNCS*, pages 32–50. Springer, 2003.
26. K. Kurosawa and Y. Desmedt. Optimum traitor tracing and asymmetric schemes. In *Proc. of EUROCRYPT*, LNCS, pages 145–157. Springer, 1998.
27. A. Langlois and D. Stehlé. Hardness of decision (R)LWE for any modulus, 2012. Available at `http://perso.ens-lyon.fr/damien.stehle/LWE4ALLQ.html`.
28. V. Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In *Proc. of ASIACRYPT*, volume 5912 of *LNCS*, pages 598–616. Springer, 2009.
29. V. Lyubashevsky and D. Micciancio. Generalized compact knapsacks are collision resistant. In *Proc. of ICALP*, volume 4052 of *LNCS*, pages 144–155. Springer, 2006.
30. V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In *Proc. of EUROCRYPT*, volume 6110 of *LNCS*, pages 1–23. Springer, 2010.
31. V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings, 2011. Draft for the extended version of [30], dated 01/02/2011.
32. D. Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Comput. Complexity*, 16(4):365–411, 2007.
33. D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput*, 37(1):267–302, 2007.
34. 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.
35. M. Naor and B. Pinkas. Efficient trace and revoke schemes. In *Proc. of Financial Cryptography*, volume 1962 of *LNCS*, pages 1–20. Springer, 2000.
36. T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Proc. of CRYPTO*, volume 576 of *LNCS*, pages 129–140. Springer, 1991.
37. C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *Proc. of STOC*, pages 333–342. ACM, 2009.
38. C. Peikert and A. Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *Proc. of TCC*, volume 3876 of *LNCS*, pages 145–166. Springer, 2006.
39. C. Peikert, A. Shelat, and A. Smith. Lower bounds for collusion-secure fingerprinting. In *Proc. of SODA*, pages 472–479, 2003.
40. C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In *Proc. of STOC*, pages 187–196. ACM, 2008.
41. B. Pfitzmann. Trials of traced traitors. In *Information Hiding*, volume 1174 of *LNCS*, pages 49–64. Springer, 1996.
42. B. Pfitzmann and M. Waidner. Asymmetric fingerprinting for larger collusions. In *ACM Conference on Computer and Communications Security*, pages 151–160, 1997.
43. 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.

44. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proc. of STOC*, pages 84–93. ACM, 2005.
45. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009.
46. O. Regev. The learning with errors problem, 2010. Invited survey in CCC 2010, available at `http://www.cs.tau.ac.il/~odedr/`.
47. C.-P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.
48. 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.
49. D. Stehlé and R. Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. In *Proc. of EUROCRYPT*, volume 6632 of *LNCS*, pages 27–47. Springer, 2011.
50. D. Stehlé, R. Steinfeld, K. Tanaka, and K. Xagawa. Efficient public key encryption based on ideal lattices. In *Proc. of ASIACRYPT*, volume 5912 of *LNCS*, pages 617–635. Springer, 2009.
51. 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.
52. 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.
53. G. Tardos. Optimal probabilistic fingerprint codes. *J. ACM*, 55(2), 2008.

Proof of Lemma 2

Proof. We use the transferance bound $\lambda_{mn}(\Lambda^{\perp}(A))\lambda_1(L) \leq mn$, where L is the dual lattice of $\Lambda^{\perp}(A)$. By adapting [49, Le. 7], it can be seen that $L = \frac{1}{q}L_q(\boldsymbol{A}^{\times})$ where $\boldsymbol{A}^{\times}$ is the matrix obtained by replacing each entry $a_{i,j}(x) \in R_q$ of $\boldsymbol{A}$ by $a_{i,j}(x^{-1})$, and

$$L_q(\boldsymbol{B}) = \left\{\boldsymbol{y} \in R^m : \exists \boldsymbol{s} \in R_q^d, \boldsymbol{B}\boldsymbol{s} = \boldsymbol{y} \bmod q\right\}, \quad \text{for any} \quad \boldsymbol{B} \in R_q^{m\times d}.$$

As the map $a(x) \mapsto a(x^{-1})$ is a bijection, it suffices to prove that:

$$\Pr_{A\hookleftarrow U(R_q^{m\times k})}\left[\lambda_1^{\infty}(L) \geq \frac{1}{8\sqrt{n}}q^{1-\frac{k}{m}}\right] \geq 1 - \left(\frac{1}{2\sqrt{n}}\right)^{nk}.$$

We obtain this result by generalizing the proof of [49, Le. 8]. By the union bound, the probability that $L_q(\boldsymbol{A})$ contains a non-zero vector of infinity norm $\leq B := \frac{1}{8\sqrt{n}}q^{1-\frac{k}{m}}$ is bounded from above by:

$$\sum_{\substack{\boldsymbol{t}\in R_q^m \\ 0<\|\boldsymbol{t}\|_{\infty}\leq B}} \sum_{\boldsymbol{s}\in R_q^k} \Pr_{\boldsymbol{A}\hookleftarrow U(R_q^{m\times k})}[\boldsymbol{A}\boldsymbol{s} = \boldsymbol{t}] \quad = \quad \sum_{\substack{\boldsymbol{t}\in R_q^m \\ 0<\|\boldsymbol{t}\|_{\infty}\leq B}} \sum_{\boldsymbol{s}\in R_q^k} \prod_{i\leq m} \Pr_{\boldsymbol{a}\hookleftarrow U(R_q^k)}[\langle \boldsymbol{a}, \boldsymbol{s}\rangle = t_i].$$

We now consider the probability (over the randomness of $\boldsymbol{a}$) that $\langle \boldsymbol{a}, \boldsymbol{s}\rangle = t_i$. For this purpose, we consider the decomposition of R_q as a Cartesian product of finite fields. If $x^n + 1 = \prod_j \Phi_j \bmod q$ with irreducible polynomials Φ_j, then these Φ_j's have a common degree δ dividing n. Then we have $R_q \simeq (\mathbb{F}_{q^{\delta}})^{\frac{n}{\delta}}$, where $\mathbb{F}_{q^{\delta}}$ is the field with q^{δ} elements. This ring isomorphism can be made explicit: It is given by the Chinese Remainder Theorem map $x \mapsto (x \bmod \Phi_1, \ldots, x \bmod \Phi_{n/\delta})$. Now, the equality $\langle \boldsymbol{a}, \boldsymbol{s}\rangle = t_i$ holds if and only if it holds over all CRT components. Wlog we consider Φ_1. If t_i and all the coordinates of $\boldsymbol{s}$ are zero modulo Φ_1, then the probability that $\langle \boldsymbol{a}, \boldsymbol{s}\rangle = t_i \bmod \Phi_1$ is 1. Otherwise, if t_i or some coordinate of $\boldsymbol{s}$ is non-zero on that component, then the probability is $\leq q^{\delta}$. As a consequence, the probability under scope is bounded from above by:

$$\sum_{0\leq j\leq n/\delta} \sum_{\substack{h=\prod_{i\in S'}\Phi_i \\ S'\subseteq S \\ |S'|=j}} \sum_{\substack{\boldsymbol{s}\in R_q^k \\ \forall i,h|s_i}} \sum_{\substack{\boldsymbol{t}\in R_q^m \\ 0<\|\boldsymbol{t}\|_{\infty}\leq B \\ \forall i,h|t_i}} q^{m(j\delta-n)} \quad \leq \quad \sum_{0\leq j\leq n/\delta} \sum_{\substack{h=\prod_{i\in S'}\Phi_i \\ S'\subseteq S \\ |S'|=j}} \sum_{\substack{\boldsymbol{t}\in R_q^m \\ \|\boldsymbol{t}\|_{\infty}\leq B \\ \forall i,h|t_i}} q^{(m-k)(j\delta-n)}.$$

The rest of the proof is as in [49]. □