Efficient Public Trace and Revoke from Standard Assumptions

Shweta Agrawal[1], Sanjay Bhattacherjee[2], Duong Hieu Phan[3], Damien Stehlé[4], and Shota Yamada[5]

[1] IIT Madras, India
[2] Turing Lab, ISI Kolkata, India
[3] XLIM (U. Limoges, CNRS), France
[4] ENS de Lyon, laboratoire LIP (U. Lyon, CNRS, ENSL, INRIA, UCBL), France
[5] National Institute of Advanced Industrial Science and Technology (AIST), Japan

Abstract. We provide efficient constructions for trace-and-revoke systems with public traceability in the black-box confirmation model. Our constructions achieve adaptive security, are based on standard assumptions and achieve significant efficiency gains compared to previous constructions.

Our constructions rely on a generic transformation from inner product functional encryption (IPFE) schemes to trace-and-revoke systems. Our transformation requires the underlying IPFE scheme to only satisfy a very weak notion of security – the attacker may only request a bounded number of *random* keys – in contrast to the standard notion of security where she may request an unbounded number of arbitrarily chosen keys. We exploit the much weaker security model to provide a new construction for bounded collusion and random key IPFE from the learning with errors assumption (LWE), which enjoys improved efficiency compared to the scheme of Agrawal *et al.* [CRYPTO'16].

Together with IPFE schemes from Agrawal *et al.*, we obtain trace and revoke from LWE, Decision Diffie Hellman and Decision Quadratic Residuosity.

Keywords. Inner-product functional encryption, Trace-and-revoke, Public traceability.

1 Introduction

A traitor tracing system [13] is a multi-receiver encryption system, which aids content distributors in identifying malicious receivers that construct pirate decryption boxes. In more detail, data is encrypted under some public key pk and each legitimate user of the system is provided a secret key sk_i that allows her to decrypt the content. Since nothing prevents a user from making copies of her key and selling them for profit, traitor tracing systems provide the following security guarantee to deter such behavior: if a coalition of users pool together their keys and construct a pirate decoder box which is capable of decrypting the ciphertext, then there is an efficient "trace" algorithm which, given access to any such decoder, outputs the identity of at least one guilty user.

An orthogonal functionality is that of broadcast encryption [15], where the content provider encrypts data to some subset S of users. Functionality requires that any user in S be able decrypt the content and security posits that no collusion of users outside S can do so. *Trace-and-revoke* systems combine these two functionalities – when the system is deployed, the content is encrypted to all users on the channel. However, if copyright infringement occurs, then tracing is used to detect the malicious users, or "traitors", and future content is encrypted using broadcast encryption to all users except the traitors.

Trace-and-revoke systems have been studied extensively [32, 31, 14, 25, 37] and are notoriously hard to construct (please see [11] for a detailed discussion). A desirable attribute for trace-and-revoke systems is *public traceability*, which means that the tracing algorithm does not require any additional secrets. Due to this property, the overall system remains secure even if the tracing party is compromised. Moreover, the tracing capability can be outsourced to an untrusted party in this setting.

To the best of our knowledge, algebraic trace-and-revoke systems with public traceability have only been achieved by Boneh and Waters [11], and quite recently by Nishimaki, Wichs and Zhandry (NWZ) [34]. The Boneh-Waters construction is quite powerful in that it supports malicious collusions of unbounded size but its ciphertexts are very large (their size grows proportionally to $\sqrt{N}$, where N is the total number of users) and the scheme relies on pairing groups of composite order. To achieve a ciphertext size that does not depend on the total number of users in the system, we consider the bounded collusion model, where the number of possible traitors is a priori bounded by some t that is polynomial in the security parameter λ. The bounded collusion model is quite standard in traitor tracing schemes and has received significant attention; however, until the work of Nishimaki *et al.* (NWZ) [34], all known schemes in this model [9, 20, 29] support either revocation or public traceability but not both.

Recently, Nishimaki *et al.* (NWZ) [34] provided a generic construction for traitor tracing systems from functional encryption schemes. Functional encryption [39, 10] is a generalization of public key encryption which allows fine grained access to encrypted data. We note that the strongest constructions in [34] are based on the existence of indistinguishability obfuscation [7], for which we do not at present have any candidate construction based on well established hardness assumptions. Since our focus is on efficient constructions based on well established hardness assumptions, we do not consider these in this work. One may also instantiate the NWZ compiler with a bounded collusion functional encryption scheme which can be based on standard assumptions such as the existence of public key encryption [18] or subexponential time hardness of learning with errors (LWE) [17, 5]. For trace and revoke, this results in a construction that supports public black box traceability and adaptive security in addition to anonymity of honest users and an exponential size universe of identities.

However, the generic nature of their construction results in loss of concrete efficiency. For instance, when based on the bounded collusion FE of [18], the resulting scheme has a ciphertext size which grows at least as $O(r+t)^5\mathcal{P}oly(\lambda))$ where r is the maximum size of the list of revoked users and t the maximum coalition size (please see Appendix A for an explanation of the bound). By relying on learning with errors, this blowup can be improved to $O((r+t)^4\mathcal{P}oly(\lambda))$ but at the cost of relying on heavy machinery such as attribute based encryption [19] and fully homomorphic encryption [17]. Additionally, this construction must also rely on complexity leveraging for adaptive security and learning with errors with subexponential error rates. The bounded collusion FE of [5] leads to better asymptotic bounds $O(r+t)^3\mathcal{P}oly(\lambda))$ but suffers from large polynomial factors which hurt concrete efficiency.

Our Approach. In this work, we revisit the connection between functional encryption and trace-and-revoke systems and observe that the notion of FE required for bounded collusion trace-and-revoke schemes is significantly weaker than that considered by [34]. To begin, we show that the functionality required from the underlying functional encryption scheme may be significantly weakened; rather than FE for polynomial sized circuits,[1] we show that *inner product* functional encryption (IPFE) [1, 4] suffices. Efficient constructions for IPFE satisfying adaptive security are available [4], which imply

[1] More accurately, the circuits required by the NWZ compiler are relatively simple, but ones for which we do not know any better FE constructions than the general case.

trace-and-revoke systems which are significantly simpler and more efficient than those implied by [34]. We further improve our constructions by observing that for the application of trace and revoke, the underlying IPFE schemes must be secure in a much weaker security model than full fledged IPFE: the adversary may be restricted to only make a bounded number of key queries, and only key queries for *randomly* chosen vectors. We exploit the much weaker security model to provide new constructions for bounded collusion and random key IPFE from LWE and Decision Quadratic Residuosity (DCR), which enjoy substantial benefits over using those of [4] in terms of parameter sizes. The improvement is greatest for the LWE construction, as the LWE modulus can be slightly super-polynomial rather than subexponential, itself allowing to choose a smaller LWE dimension.

Our Results. We construct efficient trace-and-revoke systems from standard assumptions. Our schemes support public, black-box traceability and achieve the strongest notion of adaptive security as defined by [11]. Our construction is generic and leverages recent constructions of modular inner product functional encryption (IPFE) [1, 4]. Moreover, by targeting the weak security game required by our application, we obtain more efficient versions of IPFE schemes that suffice for our purposes. While [34] achieves trace-and-revoke in the strong security model under the existence of public-key encryption, our approach leads to significantly more efficient schemes under the DCR, LWE and DDH assumptions. In particular, we achieve ciphertext and key sizes that are *linear* in the sum of revoked list size r and maximum coalition size t. Our DDH-based construction achieves ciphertext and key sizes $O((r+t)\lambda)$, our DCR-based construction achieves ciphertext and key sizes $\widetilde{O}((r+t)\lambda^3)$, while our LWE-based construction has ciphertext size $\widetilde{O}(r+t+\lambda)$ and key size $\widetilde{O}((r+t+\lambda)\lambda)$. We note that our security definition considers the strongest notion of "usefulness" [11] of the pirate decoder, which is not satisfied by most other constructions. Indeed some schemes [32, 14] are actually insecure in this strong game (see Appendix A for a detailed discussion). Finally, we give a DDH-based traitor tracing construction (without revocation) that supports encryption of k messages with ciphertext and key sizes $O((k+t)\lambda)$. This improves ciphertext expansion over the trace-and-revoke construction, as the plaintext messages are binary.

Our Techniques. Let $\mathcal{FE} = (\mathcal{FE}.\mathsf{Setup}, \mathcal{FE}.\mathsf{KeyGen}, \mathcal{FE}.\mathsf{Enc}, \mathcal{FE}.\mathsf{Dec})$ be a functional encryption scheme for the inner-product functionality over $\mathbb{Z}_p^\ell$. Recall the inner product functionality: the ciphertext encodes a vector $\mathbf{v} \in \mathbb{Z}_p^\ell$, the secret key encodes a vector $\mathbf{x} \in \mathbb{Z}_p^\ell$ and decryption recovers the inner product $\langle \mathbf{x}, \mathbf{v} \rangle \bmod p$.

To construct a trace-and-revoke scheme, we proceed as follows. At the time of key generation, a user id is first assigned a uniformly sampled vector $\mathbf{x}_{\mathsf{id}} \in \mathbb{Z}_p^\ell$ and the entry $p_{\mathsf{id}} = (\mathsf{id}, \mathbf{x}_{\mathsf{id}})$ is stored in the public directory pd for full public traceability. We may consider revocation and tracing as two distinct functionalities that need to be combined so that neither interferes with the security properties of the other. We employ two different techniques to implement these functionalities.

To revoke a set $\mathcal{R}$ of users with $|\mathcal{R}| \leq r$, we first deterministically compute a vector $\mathbf{v}_{\mathcal{R}} \in \mathbb{Z}_p^\ell$ such that for all $\mathsf{id} \in \mathcal{R}$, we have $\langle \mathbf{x}_{\mathsf{id}}, \mathbf{v}_{\mathcal{R}} \rangle = 0$ (modulo p). Note that this can be implemented only if $r < \ell$. At the same time, for a user $\mathsf{id} \notin \mathcal{R}$, the probability that $\langle \mathbf{x}_{\mathsf{id}}, \mathbf{v}_{\mathcal{R}} \rangle = 0$ must be negligible, as otherwise it would de facto be handled as a revoked user. To guarantee this, we require that p is $\lambda^{\omega(1)}$. Since we choose $\mathbf{x}_{\mathsf{id}}$ uniformly random, we have $\langle \mathbf{x}_{\mathsf{id}}, \mathbf{v}_{\mathcal{R}} \rangle \neq 0$ for $\mathsf{id} \notin \mathcal{R}$ with overwhelming probability.

Using the underlying $\mathcal{FE}$ scheme, we would like to encrypt the message $m \in \mathbb{Z}_p^\star$ such that the users in the set $\mathcal{R}$ are not able to decrypt the message, but users not in $\mathcal{R}$ are able to decrypt. We achieve it as follows:

$$C = (\mathcal{FE}.\mathsf{Enc}(\mathsf{pk}, m \cdot \mathbf{v}_{\mathcal{R}}), \mathcal{R}) = (C_1, C_2).$$

Here the operation $\cdot$ denotes the scalar multiplication of each component of $\mathbf{v}_{\mathcal{R}}$ with m. To decrypt, the user id with the vector $\mathbf{x}_{\mathsf{id}}$ and the $\mathcal{FE}$ secret key $sk_{\mathbf{x}_{\mathsf{id}}}$ proceeds as follows:

(a) Compute $\mathbf{v}_{\mathcal{R}}$ from $\mathcal{R}$ and abort if $\langle \mathbf{x}_{\mathsf{id}}, \mathbf{v}_{\mathcal{R}} \rangle = 0$.
(b) If $\langle \mathbf{x}_{\mathsf{id}}, \mathbf{v}_{\mathcal{R}} \rangle \neq 0$, compute

$$\frac{\mathcal{FE}.\mathsf{Dec}(\mathsf{sk}_{\mathbf{x}_{\mathsf{id}}}, C_1)}{\langle \mathbf{x}_{\mathsf{id}}, \mathbf{v}_{\mathcal{R}} \rangle} = \frac{\langle \mathbf{x}_{\mathsf{id}}, m \cdot \mathbf{v}_{\mathcal{R}} \rangle}{\langle \mathbf{x}_{\mathsf{id}}, \mathbf{v}_{\mathcal{R}} \rangle} = m.$$

A non-revoked user will be able to correctly decrypt this ciphertext with overwhelming probability. On the other hand, a revoked user cannot implement Step (b).

We now consider the (public) tracing procedure. We will show that given an oracle access to a pirate decoder $\mathcal{D}$ and a set $\mathcal{S} = \{\mathsf{id}_1, \mathsf{id}_2, \ldots, \}$ of suspected traitors with $|\mathcal{S}| \leq t$, it is possible to find an identity id in the set $\mathcal{T}$ of traitors, as long as $\mathcal{T} \subseteq \mathcal{S}$. Here, we assume $\mathcal{R} \cap \mathcal{S} = \emptyset$ for simplicity.

Given a pirate decoder $\mathcal{D}$, our tracing algorithm first finds a pair of messages m and m' such that $\mathcal{D}$ can distinguish the encryption of m and m' with noticeable probability. As we will show in the main body, such a pair can be found efficiently. Then, the tracing algorithm proceeds as follows. Let us consider a subset of suspect traitors $\mathcal{S}_i = \{\mathsf{id}_i, \mathsf{id}_{i+1}, \ldots\}$ for $i = 1, \ldots, |\mathcal{S}| + 1$. We then generate a probe ciphertext $C^{\mathcal{S}_i}$ associated to $\mathcal{S}_i$ with the following properties:

- The distribution of $C^{\mathcal{S}}$ corresponds to the normal encryption of m.
- The distribution of $C^{\emptyset}$ corresponds to the normal encryption of m'.
- The probes $C^{\mathcal{S}_{i-1}}$ and $C^{\mathcal{S}_i}$ are indistinguishable without a secret key for id_{i-1}.

The tracing algorithm then estimates the distinguishing advantage of the decoder $\mathcal{D}$ for $C^{\mathcal{S}_{i-1}}$ and $C^{\mathcal{S}_i}$ for all $i \in \{2, \ldots, |\mathcal{S}| + 1\}$.It outputs the identity id_{i-1} of the user that is excluded from $\mathcal{S}_{i-1}$ to get $\mathcal{S}_i$ such that the distinguishing advantage between them is non-negligible.

We prove that the tracing algorithm always outputs some user in $\mathcal{T}$. To see this, we first observe that by the first and second properties above, the decoder $\mathcal{D}$ distinguishes $C^{\mathcal{S}_1} = C^{\mathcal{S}}$ and $C^{\mathcal{S}_{|\mathcal{S}|+1}} = C^{\emptyset}$ with non-negligible advantage. Therefore, by the triangle inequality, there exists at least one index i such that $\mathcal{D}$ distinguishes $C^{\mathcal{S}_{i-1}}$ and $C^{\mathcal{S}_i}$ with non-negligible advantage. By the third property above, the identity id_{i-1} indeed corresponds to a traitor.

The above idea is implemented using inner product functional encryption. To create the probe ciphertext, we first set $\mathbf{v}_{\mathcal{S}} \in \mathbb{Z}_p^\ell$ as follows: If $i = 1$, we set $\mathbf{v}_{\mathcal{S}} = \mathbf{0}$; If $i = |\mathcal{S}| + 1$, we set $\mathbf{v}_{\mathcal{S}_i} = (m' - m) \cdot \mathbf{v}_{\mathcal{R}}$ where $\mathbf{v}_{\mathcal{R}}$ is chosen as in the ordinary encryption algorithm; Otherwise, we set $\mathbf{v}_{\mathcal{S}_i}$ so that

- $\langle \mathbf{x}_{\mathsf{id}}, \mathbf{v}_{\mathcal{S}_i} \rangle = 0$ for every $\mathsf{id} \in \mathcal{S}_i \cup \mathcal{R}$,
- $\langle \mathbf{x}_{\mathsf{id}}, \mathbf{v}_{\mathcal{S}_i} \rangle = (m' - m) \cdot \langle \mathbf{x}_{\mathsf{id}}, \mathbf{v}_{\mathcal{R}} \rangle$ for every $\mathsf{id} \in \mathcal{S}_1 \backslash \mathcal{S}_i$.

Note that this can be implemented only if $r+t<\ell$. We then set the probe ciphertext as follows:

$$C^{\mathcal{S}_i}=(C_1,C_2)=(\mathcal{FE}.\mathsf{Enc}(\mathsf{pk},\mathbf{v}_{\mathcal{S}_i}+m\cdot\mathbf{v}_{\mathcal{R}}),\mathcal{R})\,.$$

We will show that by setting the probe ciphertext for tracing as above, we can satisfy the three requirements. By construction, the first and the second requirements are satisfied. To see the third property, we consider the decryption result of the ciphertext using a secret key $\mathsf{sk}_{\mathbf{x}_{\mathsf{id}}}$ for id. We have

$$\frac{\mathcal{FE}.\mathsf{Dec}(\mathsf{sk}_{\mathbf{x}_{\mathsf{id}}},C_1)}{\langle\mathbf{x}_{\mathsf{id}},\mathbf{v}_{\mathcal{R}}\rangle}=\frac{\langle\mathbf{x}_{\mathsf{id}},\mathbf{v}_{\mathcal{S}_i}+m\cdot\mathbf{v}_{\mathcal{R}}\rangle}{\langle\mathbf{x}_{\mathsf{id}},\mathbf{v}_{\mathcal{R}}\rangle}=\frac{\langle\mathbf{x}_{\mathsf{id}},\mathbf{v}_{\mathcal{S}_i}\rangle}{\langle\mathbf{x}_{\mathsf{id}},\mathbf{v}_{\mathcal{R}}\rangle}+m.$$

Therefore, the decryption result of the probe ciphertext $C^{\mathcal{S}_i}$ is m if $\mathsf{id}\in\mathcal{S}_i$ and m' if $\mathsf{id}\in\mathcal{S}\backslash\mathcal{S}_i$. Then we observe that the decryption results of $C^{\mathcal{S}_i}$ and $C^{\mathcal{S}_{i-1}}$ are the same, as long as we use a secret key for $\mathsf{id}\in\mathcal{S}\cup\mathcal{R}$ with $\mathsf{id}\neq\mathsf{id}_{i-1}$. By the security property of inner product functional encryption, this implies that any coalition of users $\subseteq\mathcal{S}$ cannot distinguish two ciphertexts without having $\mathsf{sk}_{\mathbf{x}_{\mathsf{id}_{i-1}}}$. Namely, the third requirement regarding the probe ciphertext also holds.

Our LWE-based IPFE. Here, we give the overview of our direct construction of LWE-based IPFE scheme, which enjoys improved efficiency compared to [4]. Let ℓ and p be the dimension and modulus of the space on which inner-products are taken. Furthermore, let $q=p^k$ be the LWE modulus, where k is some integer. In our scheme, the master secret key is $\mathbf{Z}\in\mathbb{Z}^{\ell\times n}$, which is chosen from a Gaussian distribution with standard deviation σ. The public key is of the form $\mathsf{pk}=(\mathbf{A}\in\mathbb{Z}_q^{m\times n},\mathbf{U}=\mathbf{ZA}\in\mathbb{Z}_q^{\ell\times n})$. To generate a secret key for the vector $\mathbf{x}\in\mathbb{Z}_p^{\ell}$, we first pick a vector $\bar{\mathbf{x}}\in\mathbb{Z}^{\ell}$ from a short Gaussian distribution over $\mathbb{Z}^{\ell}$ conditioned on $\bar{\mathbf{x}}\equiv\mathbf{x}\bmod p$. Then, the secret key is set as $\mathsf{sk}_{\mathbf{x}}=(\bar{\mathbf{x}}^t,\bar{\mathbf{x}}^t\cdot\mathbf{Z})$. One may wonder why do we set $\bar{\mathbf{x}}$ like this instead of just setting $\bar{\mathbf{x}}=\mathbf{x}$. This is because we will use some nice properties of the Gaussian distribution in our security proof, which will be explained later. The ciphertext for a vector $\mathbf{y}\in\mathbb{Z}_p^{\ell}$ is of the form $(\mathbf{c}_0\approx\mathbf{As},\mathbf{c}_1\approx\mathbf{Us}+p^{k-1}\cdot\mathbf{y})$ where $\mathbf{x}\approx\mathbf{y}$ means that $\|\mathbf{x}-\mathbf{y}\|$ is small.

Here, we skip the explanation of the decryption algorithm and directly go to the intuition for the security proof. We first observe that since all entries of $\mathbf{Z}$ are small, $\mathbf{c}_1\approx\mathbf{ZAs}\approx\mathbf{Zc}_0$. Given this observation, we can change the distribution of the ciphertext as $\mathbf{c}_0$ being a random vector $\mathbf{u}\hookleftarrow\mathbb{Z}_q^{\ell}$ and $\mathbf{c}_1\approx\mathbf{Zu}+p^{k-1}\cdot\mathbf{y}$ without being detected by the adversary, assuming the LWE assumption.

The main difficulty in the proof is in showing that $\mathbf{c}_1\approx\mathbf{Zu}+p^{k-1}\mathbf{y}$ does not leak any information *more than necessary.* Note that $\mathbf{c}_1$ *does* leak some information. Namely, given a secret key $\mathsf{sk}_{\mathbf{x}}$ for $\mathbf{x}$, we can still decrypt the modified ciphertext to obtain $\langle\mathbf{x},\mathbf{y}\rangle\bmod p$. What we have to prove is that the ciphertext does not leak any information of $\langle\mathbf{x}',\mathbf{y}\rangle\bmod p$ for all $\mathbf{x}'\notin\mathrm{Span}_{\mathbb{Z}_p}(\{\mathbf{x}_i\}_{i\in[L]})$, where L is the number of key queries and $\{\mathbf{x}_1,\ldots,\mathbf{x}_L\}$ is the set of vectors for which the adversary is given corresponding secret keys.

This will be shown by an information theoretic argument using the fact that certain amount of information on $\mathbf{Z}$ is hidden from the adversary. In particular, we explain that an attempt to obtain any information of $\langle\mathbf{x}',\mathbf{y}\rangle\bmod p$ by computing $\langle\mathbf{x}',\mathbf{c}_1\rangle\approx\mathbf{x}'^t\mathbf{Zu}+p^{k-1}\cdot\langle\mathbf{x}',\mathbf{y}\rangle\bmod q$ fails because $\mathbf{x}'^t\mathbf{Z}$ retains sufficiently high min-entropy and thus $\mathbf{x}'^t\mathbf{Zu}$ is uniformly random modulo q by the leftover hash lemma.

To see this, let $\mathbf{X}_{top}\in\mathbb{Z}^{L\times\ell}$ be the matrix which is obtained by vertically concatenating $\{\bar{\mathbf{x}}_i\in\mathbb{Z}^{\ell}\}_{i\in L}$. Via secret keys, the adversary learns the value of $\mathbf{X}_{top}\mathbf{Z}$. Let us ignore the additional leakage on $\mathbf{Z}$ from the public key in this overview. Note that in $\mathbf{X}_{top}\mathbf{Z}$, the matrix $\mathbf{X}_{top}$ acts in parallel on the columns of $\mathbf{Z}$. We can hence restrict ourselves to the distribution of $\mathbf{z}_i$ conditioned on $\mathbf{b}_i := \mathbf{X}_{top}\mathbf{z}_i$. It can be seen that $\mathbf{z}_i$ is distributed on the shifted kernel lattice Λ, which is defined as

$$\Lambda = \{\mathbf{v} \in \mathbb{Z}^m \ : \ \mathbf{X}_{top} \cdot \mathbf{v} = 0\}.$$

If the standard deviation σ is sufficiently large (i.e., larger than the smoothing parameter of Λ), the vector $\mathbf{z}_i$ is distributed as if it were chosen from the continuous Gaussian distribution. In particular, it spreads all directions under the only constraint that $\mathbf{X}_{top}\mathbf{z}_i = \mathbf{b}_i$, and thus $\langle \mathbf{x}', \mathbf{z}_i \rangle$ has sufficiently high entropy, which allows us to conclude. In [4], the equivalent of $\mathbf{X}_{top}$ in their proof is arbitrarily chosen by the adversary and $\ell = L + 1$. This results in exponentially large smoothing parameter for corresponding Λ. Therefore, they have to take σ exponentially large, which is exactly the source of the inefficiency in their scheme. In our case, the matrix $\mathbf{X}_{top}$ is chosen uniformly at random from a small-width Gaussian distribution. (Recall that in our weakened security definition, the adversary does not have control over $\mathbf{x}_i$.) Furthermore, we set ℓ large compared to L. We can then invoke the result of [3], which says that the smoothing parameter of Λ corresponding to such $\mathbf{X}_{top}$ is small. This allows us to choose σ much smaller and significantly improve the efficiency.

Organization of the paper. The remainder of the paper is organized as follows. In Section 2, we provide definitions and preliminaries required for our work. In Section 3, we provide our generic construction of trace-and-revoke systems from inner product functional encryption. In Section 4, we provide our new construction of bounded collusion IPFE from LWE and in Section 5 we provide concrete instantiations of trace-and-revoke systems from the DDH and DCR assumptions. In Appendix B we provide a generic transformation from an inner product functional encryption scheme to a traitor tracing scheme that supports multi-message encryption.

2 Definitions and Preliminaries

Notation. The set $\{1, \ldots, n\}$ of natural numbers is denoted by $[n]$. A set is denoted by an uppercase letter. The cardinality of a set X is denoted as $|X|$. If X is finite, we let $U(X)$ denote the uniform distribution over X, and we may write $x \hookleftarrow X$ to refer to x being sampled from $U(X)$. Vectors will be denoted by bold letters. By default, we treat a vector as a column vector. For two vectors $\mathbf{x}$ and $\mathbf{y}$, we let $\langle \mathbf{x}, \mathbf{y} \rangle$ denote the canonical inner product between them and $(\mathbf{x} \| \mathbf{y})$ denote the vertical concatenation of them. For a positive integer N, we let $\mathbb{Z}_N$ denote the ring of integers with addition and multiplication modulo N. The set of all functions that run in polynomial time is denoted by $\mathcal{P}oly(\cdot)$.

In our scheme descriptions, a user's identifying information is denoted by id. A set of users is thus represented by a set of their respective identifying information. A set of users is denoted by an uppercase calligraphic letter. The set of revoked users is denoted by $\mathcal{R}$. The set of traitors is denoted by $\mathcal{T}$ and the set of users that are suspected to be traitors is denoted by $\mathcal{S}$.

In this section, we recall the notions of trace-and-revoke systems and inner product functional encryption.

2.1 Trace-and-Revoke Systems

In a public key traitor tracing encryption scheme, there is a single public key for encryption and many users with decryption capabilities, each having its own unique secret key. Additionally, the encryption scheme provides a feature to identify at least one user from a coalition of malicious users (traitors) that built an unauthorized decryption device $\mathcal{D}$. Let $\mathcal{T}$ be the set of traitors and we assume that the size $|\mathcal{T}|$ of the traitor coalition is at most t. The tracing algorithm aims at disclosing the identity of at least one user from the set $\mathcal{T}$ of traitors.

In [9], the minimal black-box access model was considered where the tracing procedure has access to the pirate decryption device $\mathcal{D}$ only through an oracle $\mathcal{O}^{\mathcal{D}}$. The oracle $\mathcal{O}^{\mathcal{D}}$ takes as input any message-ciphertext pair (M, C) and returns 1 if $\mathcal{D}(C) = M$ and 0 otherwise. Hence, it only tells whether the decoder decrypts C to M or not. If the decoder fails to decrypt correctly, the tracing algorithm knows nothing about the decrypted value returned by the decoder. A practical example supporting this assumption is that a pirated media player will only indicate if it is able to play some encrypted media and nothing more about the results of his attempts of decryption.

The decryption device $\mathcal{D}$ is assumed to decrypt correctly with significant probability all messages that have been properly encrypted, as otherwise the decryption device is not very useful. Let $\mathcal{R}$ be any set of revoked users, of cardinality $\leq r$. Let the message m be sampled uniformly at random from the message space $\mathcal{M}$ and let $C^{(\mathcal{R})}$ be the output of the encryption algorithm $\mathtt{Enc}$ using the public encryption key pk and $\mathcal{R}$ as the set of revoked users. With $C^{(\mathcal{R})}$ as input, the device $\mathcal{D}$ outputs m with probability significantly more than $1/|\mathcal{M}|$:

$$\Pr_{\substack{m \hookleftarrow U(\mathcal{M}) \\ C^{(\mathcal{R})} \hookleftarrow \mathtt{Enc}(\mathsf{pk}, \mathcal{R}, m)}} \left[\mathcal{O}^{\mathcal{D}}(C^{(\mathcal{R})}, m) = 1 \right] \geq \frac{1}{|\mathcal{M}|} + \frac{1}{\lambda^c}, \tag{1}$$

for some constant $c > 0$.[2] The probability of decryption for a decoder $\mathcal{D}$ can be estimated by repeatedly querying the oracle $\mathcal{O}^{\mathcal{D}}$ with plaintext-ciphertext pairs, using Hoeffding's inequality. Alternatively, we may force the correct decryption probability to be non-negligibly close to 1, by using an all-or-nothing 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].

We let the identity space ID and the message space $\mathcal{M}$ be implicit arguments to the setup algorithm below. We let the secret key space $\mathcal{K}$ and the ciphertext space $\mathcal{C}$ (along with ID and $\mathcal{M}$) be implicit public parameters output by the setup algorithm.

Definition 1. A dynamic identity-based trace-and-revoke scheme (t, r)-$\mathcal{TR}$ in black-box confirmation model is a tuple $\mathcal{TR} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Trace})$ of five probabilistic polynomial-time algorithms with the following specifications.

- $\mathsf{Setup}(1^\lambda, 1^t, 1^r)$ takes as input the security parameter λ, the bound t on the size of traitor coalitions and the bound r on the number of revoked users. It out-

[2] In [34], a weaker notion of usefulness is considered (leading to a better security guarantee): the box is considered useful if it distinguishes between encryptions of two adversarially chosen plaintexts. We note that our security proof actually handles this weaker usefulness. In fact, we show in Lemma 14 that the notion of usefulness given here implies that it is possible to efficiently find two plaintexts whose ciphertext distributions can be distinguished by the decryption box. The rest of the security proof carries over in an identical way for both usefulness notions.

puts $(\mathsf{msk}, \mathsf{pk}, \mathsf{pd})$ containing the master secret key, the encryption key and the (initially empty) public directory pd.

- $\mathsf{KeyGen}(\mathsf{msk}, \mathsf{id})$ takes as input the master secret msk and an identity $\mathsf{id} \in \mathsf{ID}$ of a user, and outputs a secret key sk_{id} and some public information p_{id} for id. It also updates the public directory pd to include p_{id}.[3]
- $\mathsf{Enc}(\mathsf{pk}, \mathcal{R}, m)$ takes as input the public key pk, a set $\mathcal{R}$ of cardinality $\leq r$ which contains the p_{id} of each revoked user in pd, and a plaintext message $m \in \mathcal{M}$. It outputs a ciphertext $C \in \mathcal{C}$.
- $\mathsf{Dec}(\mathsf{sk}_{\mathsf{id}}, C)$ takes as input a secret key $\mathsf{sk}_{\mathsf{id}}$ of a user with identity id and a ciphertext $C \in \mathcal{C}$. It outputs a plaintext $m \in \mathcal{M}$.
- $\mathsf{Trace}(\mathsf{pd}, \mathcal{R}, \mathcal{S}, \mathcal{O}^{\mathcal{D}})$ is a black-box confirmation tracing algorithm that takes as input the public directory pd, a set $\mathcal{R}$ of $\leq r$ revoked users, a set $\mathcal{S}$ of $\leq t$ suspect users, and has black-box access to the pirate decoder $\mathcal{D}$ through the oracle $\mathcal{O}^{\mathcal{D}}$. It outputs an identity id or $\perp$.

The correctness requirement is that, with overwhelming probability over the randomness used by the algorithms, we have:

$$\forall m \in \mathcal{M}, \forall \mathsf{id} \in \mathsf{ID} : \mathsf{Dec}(\mathsf{sk}_{\mathsf{id}}, \mathsf{Enc}(\mathsf{pk}, \mathcal{R}, m)) = m,$$

for any set $\mathcal{R}$ of $\leq r$ revoked users and for any id such that $\mathsf{id} \notin \mathcal{R}$.

Public Traceability. It is required that, when $\mathcal{S}$ contains the set $\mathcal{T}$ of traitors who produced the pirate decoder $\mathcal{D}$, then the id output by the tracing algorithm belongs to $\mathcal{T}$. This requirement is formalized using the following game, denoted by $\mathsf{AD\text{-}TT}$, between an adversary $\mathcal{A}$ and a challenger:

- The challenger runs $\mathsf{Setup}(1^\lambda, 1^t, 1^r)$ and gives pk to $\mathcal{A}$.
- Adversary $\mathcal{A}$ may ask the challenger to add polynomially many users in the system. Adversary $\mathcal{A}$ may choose the id's of the users, but does not obtain the corresponding $\mathsf{sk}_{\mathsf{id}}$. Nevertheless, the public directory pd is updated accordingly.
- Adversary $\mathcal{A}$ is allowed to make up to t arbitrary traitor key queries. It may observe the database pd to choose its queries in an adaptive way. If it queries $\mathsf{id} \in \mathsf{ID}$ to the challenger, then:
  - If the key for id was previously generated, i.e., if p_{id} is found in the database pd, then the challenger responds with $\mathsf{sk}_{\mathsf{id}}$. The challenger records the identity query id in a list $\mathcal{T}$.
  - Otherwise (i.e., user id is a new user in the system), the challenger runs $\mathsf{KeyGen}(\mathsf{msk}, \mathsf{id})$, responds with $\mathsf{sk}_{\mathsf{id}}$ and updates the directory pd with the public information p_{id} for id. The challenger also records the identity query id in the list $\mathcal{T}$.
- Adversary $\mathcal{A}$ is allowed to (adaptively) choose a set $\mathcal{R}$ of up to r revoked users in pd. The challenger gives $\mathcal{A}$ all the corresponding $\mathsf{sk}_{\mathsf{id}}$.
- Adversary $\mathcal{A}$ finally produces a pirate decoder $\mathcal{D}$. It chooses a suspect set $\mathcal{S}$ of cardinality $\leq t$ that contains $\mathcal{T}$, and sends $\mathcal{S}$ to the challenger.
- The challenger then runs $\mathsf{Trace}(\mathsf{pd}, \mathcal{R}, \mathcal{S}, \mathcal{O}^{\mathcal{D}})$. The adversary wins if both of the following hold:
  - Equation (1) is satisfied for the set of revoked users $\mathcal{R}$ chosen by the adversary (i.e., decoder $\mathcal{D}$ is useful),

[3] We emphasize that p_{id} does not need to contain id.

– the execution of Trace outputs $\perp$ or outputs an id that does not belong to $\mathcal{T}$ with probability $\geq 1/\lambda^c$.

No probabilistic polynomial-time adversary $\mathcal{A}$ should be able to win game $\mathsf{AD\text{-}TT}$ with non-negligible probability.

Almost Public Traceability. This is the same as public traceability, except that Trace only outputs the associated information about the traitors instead of their identities, namely p_{id} instead of id. Consequently, the second winning condition of the adversary should be adapted so that it only requires the execution of Trace to output a p_{id} that does not belong to $\mathsf{pd}_{\mathcal{T}}$, which is the set of all $p_{\mathsf{id}'}$ for $\mathsf{id}' \in \mathcal{T}$.

This restriction does not change much the functionality of the tracing because, from p_{id}, the authority can immediately map back to id and the authority can still delegate the tracing procedure to untrusted parties. On the other side, this variant may be useful in practice as we do not leak the information of users in the public directory.

Traitor Tracing Scheme. A traitor tracing scheme is simply a trace-and-revoke scheme without the capacity of revoking users. It corresponds to the above definition where the revoked set is always set to be empty, in the encryption as well as in the security game.

Semantic Security. The $\mathsf{IND\text{-}CPA}$ security of a trace-and-revoke scheme $\mathcal{TR}$ is defined based on the following game.

- The challenger runs $\mathsf{Setup}(1^\lambda, 1^t, 1^r)$ and gives the produced public key pk to the adversary $\mathcal{A}$. The adversary may ask the challenger to add polynomially many users in the system.
- The adversary (adaptively) chooses a set $\mathcal{R}$ of $\leq r$ revoked users in pd. The challenger gives $\mathcal{A}$ all the $\mathsf{sk}_{\mathsf{id}}$ such that $p_{\mathsf{id}} \in \mathcal{R}$.
- The adversary then chooses two messages $m_0, m_1 \in \mathcal{M}$ of equal length and gives them to the challenger.
- The challenger samples $b \hookleftarrow \{0,1\}$ and provides $C_{m_b} \hookleftarrow \mathsf{Enc}(\mathsf{pk}, \mathcal{R}, m_b)$ to $\mathcal{A}$.
- Finally, the adversary returns its guess $b' \in \{0,1\}$ for the b chosen by the challenger. The adversary wins this game if $b = b'$.

The advantage of the adversary is defined as $\mathrm{Adv}_{\mathcal{TR},\mathcal{A}}^{\mathsf{IND\text{-}CPA}} = |\Pr[b = b'] - 1/2|$. The scheme $\mathcal{TR}$ is said semantically secure if there is no probabilistic polynomial-time adversary $\mathcal{A}$ that wins this game with non-negligible advantage.

2.2 Inner Product Functional Encryption

In this section, we define functional encryption for the functionality of inner products over $\mathbb{Z}_p$.

Definition 2. A *functional encryption* scheme $\mathcal{FE}$ for the inner product functionality over $\mathbb{Z}_p$ is a tuple $\mathcal{FE} = (\mathcal{FE}.\mathsf{Setup}, \mathcal{FE}.\mathsf{KeyGen}, \mathcal{FE}.\mathsf{Enc}, \mathcal{FE}.\mathsf{Dec})$ of four probabilistic polynomial-time algorithms with the following specifications:

- $\mathcal{FE}.\mathsf{Setup}(1^\lambda, 1^\ell)$ takes as input the security parameter λ and outputs the public key and the master secret key pair $(\mathsf{pk}, \mathsf{msk})$;
- $\mathcal{FE}.\mathsf{KeyGen}(\mathsf{msk}, \mathbf{x})$ takes as input the master secret key msk and a vector $\mathbf{x} \in \mathbb{Z}_p^\ell$ and outputs the secret key $\mathsf{sk}_{\mathbf{x}}$;

- $\mathcal{FE}.\mathsf{Enc}(\mathsf{pk}, \mathbf{y})$ takes as input the public key pk and a message $\mathbf{y} \in \mathbb{Z}_p^\ell$ and outputs the ciphertext $\mathsf{ct}_\mathbf{y}$;
- $\mathcal{FE}.\mathsf{Dec}(\mathsf{sk}_\mathbf{x}, \mathsf{ct}_\mathbf{y})$ takes as input the secret key of a user $\mathsf{sk}_\mathbf{x}$ and the ciphertext $\mathsf{ct}_\mathbf{y}$, and outputs an element from $\mathbb{Z}_p \cup \{\perp\}$.

The correctness requirement is that, with overwhelming probability over the randomness used by the algorithms, for $(\mathsf{pk}, \mathsf{msk}) \hookleftarrow \mathcal{FE}.\mathsf{Setup}(1^\lambda, 1^\ell)$ and $\forall \mathbf{x}, \mathbf{y} \in \mathbb{Z}_p^\ell$:

$$\mathcal{FE}.\mathsf{Dec}\left(\mathcal{FE}.\mathsf{KeyGen}(\mathsf{msk}, \mathbf{x}), \mathcal{FE}.\mathsf{Enc}(\mathsf{pk}, \mathbf{y})\right) = \langle \mathbf{x}, \mathbf{y} \rangle \bmod p.$$

Security of $\mathcal{FE}$. We consider security of functional encryption in the standard indistinguishability setting [10].

Definition 3. A functional encryption scheme $\mathcal{FE} = (\mathcal{FE}.\mathsf{Setup}, \mathcal{FE}.\mathsf{KeyGen}, \mathcal{FE}.\mathsf{Enc}, \mathcal{FE}.\mathsf{Dec})$ provides semantic security under chosen-plaintext attacks (or IND-CPA security) if no probabilistic polynomial-time adversary $\mathcal{A}$ has non-negligible advantage in the following game:

- The challenger runs $\mathcal{FE}.\mathsf{Setup}(1^\lambda, 1^\ell)$ and the master public key mpk is given to $\mathcal{A}$.
- The adversary adaptively makes secret key queries to the challenger. At each query, adversary $\mathcal{A}$ chooses a vector $\mathbf{x} \in \mathbb{Z}_p^\ell$ and obtains the corresponding secret key $\mathsf{sk}_\mathbf{x} \hookleftarrow \mathcal{FE}.\mathsf{KeyGen}(\mathsf{msk}, \mathbf{x})$.
- Adversary $\mathcal{A}$ chooses distinct messages $\mathbf{y}_0, \mathbf{y}_1 \in \mathbb{Z}_p^\ell$ subject to the restriction that, for every vector $\mathbf{x}$ queried in the previous step, it holds that $\langle \mathbf{x}, \mathbf{y}_0 \rangle = \langle \mathbf{x}, \mathbf{y}_1 \rangle \bmod p$ and sends them to the challenger. In response, the challenger samples $b \hookleftarrow \{0, 1\}$ and sends $\mathsf{ct}^\star \hookleftarrow \mathcal{FE}.\mathsf{Enc}(\mathsf{pk}, \mathbf{y}_b)$ to $\mathcal{A}$.
- Adversary $\mathcal{A}$ makes further secret key queries for arbitrary vectors $\mathbf{x} \in \mathbb{Z}_p^\ell$ of its choice. As before, it is required that $\langle \mathbf{x}, \mathbf{y}_0 \rangle = \langle \mathbf{x}, \mathbf{y}_1 \rangle \bmod p$ for each query $\mathbf{x}$ made by $\mathcal{A}$.
- Adversary $\mathcal{A}$ eventually outputs a bit $b' \in \{0, 1\}$ and wins if $b' = b$.

The adversary's advantage is defined as $\mathbf{Adv}_\mathcal{A}(\lambda) := |\Pr[b' = b] - 1/2|$.

The Random-Key Bounded-Collusion Model. In bounded collusion functional encryption [18], the adversary $\mathcal{A}$ is restricted to ask at most Q secret key queries for some fixed polynomial Q, which is input to the setup algorithm. Additionally, our application permits an additional weakening of the security model for inner product functional encryption: we are only required to show security against an adversary who first sees arbitrarily many *random* vectors $\mathbf{x} \hookleftarrow \mathbb{Z}_p^\ell$, requests secret keys for an adaptively chose subset of them, and does not make secret key queries after it gets the challenge ciphertext. The above definition of security against such a restricted adversary will be called Q-IND-CPA.

2.3 Lattice background

A lattice Λ is a (non-zero) discrete subgroup of $\mathbb{R}^m$. A *basis* of Λ is a linearly independent set of vectors whose $\mathbb{Z}$-span is Λ. We recall that the *smoothing parameter* of Λ is defined as

$$\eta_\varepsilon(\Lambda) = \min\Big(\sigma > 0 : \sum_{\widehat{\mathbf{b}} \in \widehat{\Lambda}} \exp(-\pi \|\widehat{\mathbf{b}}\|^2/\sigma^2) \le 1 + \varepsilon\Big),$$

where $\widehat{\Lambda} = \{\widehat{\mathbf{b}} \in \mathrm{Span}_\mathbb{R}(\Lambda) : \widehat{\mathbf{b}}^T \cdot \Lambda \subseteq \mathbb{Z}\}$ refers to the dual of Λ. Note that if $\sigma = \Omega(\sqrt{\lambda})$, we have that there exists $\varepsilon = 2^{-\Omega(\lambda)}$ such that $\sigma \ge \eta_\varepsilon(\mathbb{Z})$.

For a lattice $\varLambda \subseteq \mathbb{R}^m$, a vector $\mathbf{c} \in \mathbb{R}^m$, and an invertible $\mathbf{\Sigma} \in \mathbb{R}^{m\times m}$, we define the Gaussian distribution of parameter $\varLambda$, $\mathbf{c}$, and $\mathbf{\Sigma}$ by $D_{\varLambda,\mathbf{\Sigma},\mathbf{c}}(\mathbf{b}) \sim \rho_{\mathbf{\Sigma},\mathbf{c}}(\mathbf{b}) = \exp(-\pi\|\mathbf{\Sigma}^{-1}(\mathbf{b}-\mathbf{c})\|^2)$ for all $\mathbf{b} \in \varLambda$. When $\mathbf{\Sigma} = \sigma\mathbf{I}_m$, we simply write $D_{\varLambda,\sigma,\mathbf{c}}$. Sometimes, for convenience, we use the notation $D_{\varLambda+\mathbf{c},\mathbf{\Sigma}}$ as a shorthand for $\mathbf{c} + D_{\varLambda,\mathbf{\Sigma},-\mathbf{c}}$.

For $m \geq n$ and a rank-n matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, denote $U_{\mathbf{X}} = \{\|\mathbf{X}\mathbf{u}\| : \mathbf{u} \in \mathbb{R}^n, \|\mathbf{u}\| = 1\}$. The least singular value of $\mathbf{X}$ is then defined as $s_n(\mathbf{X}) := \inf(U_{\mathbf{X}})$ and similarly the largest singular value of $\mathbf{X}$ is $s_1(\mathbf{X}) := \sup(U_{\mathbf{X}})$. For a matrix $\mathbf{Y} \in \mathbb{R}^{n'\times m'}$ with $n' > m'$, the least singular value and the largest singular value are defined as $s_1(\mathbf{Y}) := s_1(\mathbf{Y}^t)$ and $s_{m'}(\mathbf{Y}) := s_{m'}(\mathbf{Y}^t)$ respectively.

For the rest of this section, we assume that lattices are full-rank, i.e., the dimensions of the span and the ambient space match.

Lemma 4 (Corollary 2.8 in [16]). *Let $\varLambda' \subseteq \varLambda \subseteq \mathbb{R}^m$ be two lattices with the same dimension. Let $\varepsilon \in (0, 1/2)$. Then for any $\mathbf{c} \in \mathbb{R}^m$ and any $\mathbf{\Sigma}$ such that $s_m(\mathbf{\Sigma}) \geq \eta_\varepsilon(\varLambda')$, the distribution $D_{\varLambda,\mathbf{\Sigma},\mathbf{c}} \bmod \varLambda'$ is within statistical distance 2ε from the uniform distribution over $\varLambda/\varLambda'$.*

Lemma 5 (Lemma 1 in [21]). *Let $q, \ell, m > 0$ integers and $r \geq \Omega(\sqrt{\lambda})$. Let $\mathbf{b} \in \mathbb{Z}_q^m$ be arbitrary and $\mathbf{x}$ chosen from $D_{\mathbb{Z}^m,r}$. Then for any $\mathbf{V} \in \mathbb{Z}^{\ell\times m}$ and positive real $r' > s_1(\mathbf{V})$, there exists a probabilistic polynomial-time algorithm $\mathsf{ReRand}(\mathbf{V}, \mathbf{b} + \mathbf{x}, r, r')$ that outputs $\mathbf{b}' = \mathbf{V}\mathbf{b} + \mathbf{x}' \in \mathbb{Z}_q^\ell$ where $\mathbf{x}'$ is within statistical distance $2^{-\Omega(\lambda)}$ from $D_{\mathbb{Z}^\ell,2rr'}$.*

We use the following variant of the leftover hash lemma, adapted from [30] (see also Lemma 11 in [4]).

Lemma 6 ([30]). *For $q = p^k$ for p prime and $k \geq 1$. Let $m \geq n \geq 1$. Take $\mathcal{X}$ a distribution over $\mathbb{Z}^m$. Let D_0 be a uniform distribution over $\mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^n$ and D_1 be the distribution of $(\mathbf{A}, \mathbf{A}\cdot\mathbf{x})$, where sampling $\mathbf{A} \hookleftarrow \mathbb{Z}_q^{n\times m}$ and $\mathbf{x} \hookleftarrow \mathcal{X}$. Then,*

$$\varDelta(D_0, D_1) \leq \frac{1}{2}\sqrt{\sum_{i=1}^{k} p^{i\cdot n} \cdot \Pr{}_i}.$$

where $\Pr_i$ is the collision probability of two independent samples from $(\mathcal{X} \bmod p^i)$.

The above lemma implies that if the distribution $(\mathcal{X} \bmod p)$ is within statistical distance ε from the uniform distribution over $\mathbb{Z}_p^m$, then we have

$$\varDelta(D_0, D_1) \leq \varepsilon + \sqrt{q^n/p^m}.$$

This can be seen by considering a distribution $\mathcal{X}'$ such that $(\mathcal{X} \bmod p)$ is uniform distribution over $\mathbb{Z}_p^m$ and $\varDelta(\mathcal{X}, \mathcal{X}') \leq \varepsilon$.

Lemma 7 (Special case of Lemma 8 in [4]). *There exists a universal constant $K > 1$ such that for all $m \geq 2n$, $\varepsilon > 0$ and $\sigma \geq K\eta_\varepsilon(\mathbb{Z})$, the following holds for $\mathbf{X} \hookleftarrow D_{\mathbb{Z},\sigma}^{n\times m}$:*

$$\Pr\left[\sigma\sqrt{2\pi m}/K < s_n(\mathbf{X}) \leq s_1(\mathbf{X}) < \sigma K\sqrt{2\pi m}\right] > 1 - 4m\varepsilon + O(\exp(-m/K)).$$

We will also require the following theorem, adapted from Theorem 17 in [29].

Theorem 8 ([29]). *Let n, m_1, m_2, and λ be integers satisfying $m_2 \geq m_1 > 100$ and σ_1, σ_2 be real numbers. Let us set $n' = \max\{\lambda, n\}$ and assume $n' > 100$. We also assume that they satisfy $\sigma_1 \geq \Omega(\sqrt{m_1 n' \log m_1})$, $m_1 \geq \Omega(n' \log(\sigma_1 n'))$, and $\sigma_2 \geq \Omega(n'^{5/2}\sqrt{m_1}\sigma_1^2 \log^{3/2}(m_1\sigma_1))$. Then, there exists a probabilistic polynomial-time algorithm that given n, m_1, m_2, λ (in unary), σ_1, and σ_2, returns $\mathbf{X}_1 \in \mathbb{Z}^{n\times m_1}$, $\mathbf{X}_2 \in \mathbb{Z}^{n\times m_2}$, and $\mathbf{U} \in \mathbb{Z}^{m\times m}$ with $m = m_1 + m_2$ such that:*

- *the distribution of $(\mathbf{X}_1, \mathbf{X}_2)$ is within statistical distance $2^{-\Omega(n')}$ of the distribution $D_{\mathbb{Z},\sigma_1}^{n\times m_1} \times (D_{\mathbb{Z}^{m_2},\sigma_2,\boldsymbol{\delta}_1} \times \cdots \times D_{\mathbb{Z}^{m_2},\sigma_2,\boldsymbol{\delta}_n})^t$, where $\boldsymbol{\delta}_i$ denotes the ith canonical unit vector in $\mathbb{Z}^{m_2}$ whose ith coordinate is 1 and whose remaining coordinates are 0,*
- *we have $|\det \mathbf{U}| = 1$ and $(\mathbf{X}_1|\mathbf{X}_2) \cdot \mathbf{U} = (\mathbf{I}_n|\mathbf{0})$,*
- *every column of $\mathbf{U}$ has norm $\leq O(\sqrt{n'm_1}\sigma_2)$ with probability $\geq 1 - 2^{-\Omega(n')}$.*

Three remarks are in order regarding the theorem. First, we take the transpose of the theorem in [29]. This is just for a notational convenience. Secondly, the distribution of $\mathbf{X} = (\mathbf{X}_1|\mathbf{X}_2)$ in Theorem 17 in [29] is slightly different from the above in that all entries of the first column of $\mathbf{X}$ equal to 1. As noted right after Lemma 7 in [29], the theorem still holds even with the change. Finally, in the above theorem, we introduce the statistical security parameter λ and differentiate it from the lattice dimension n, while the theorem in [29] assigns the same variable n for both. This change is introduced because we will invoke the theorem for possibly small n for which 2^{-n} is no longer negligible.

In our security analysis, we need a variant of the above theorem where $\mathbf{X}$ is chosen from a slightly different distribution and $\mathbf{U}$ need not be efficiently samplable.

Lemma 9. *Let n, m_1, m_2, m, λ, n', σ_1, σ_2 be as in Theorem 8. Then, for all but $2^{-\Omega(n')}$ probability over $(\mathbf{X}_1, \mathbf{X}_2) \in \mathbb{Z}^{n\times m_1} \times \mathbb{Z}^{n\times m_2}$ chosen from $D_{\mathbb{Z},\sigma_1}^{n\times m_1} \times D_{\mathbb{Z},\sigma_2}^{n\times m_2}$, there exists $\mathbf{U} \in \mathbb{Z}^{m\times m}$ such that $|\det \mathbf{U}| = 1$, $(\mathbf{X}_1|\mathbf{X}_2)\cdot\mathbf{U} = (\mathbf{I}_n|\mathbf{0})$, and every column of $\mathbf{U}$ has norm $\leq O(\sqrt{n'm_1}\sigma_2)$.*

To prepare for the proof of Lemma 9, we define Rényi Divergence (RD) and review its properties following [6]. For any two probability distributions P and Q such that the support of P is a subset of the support of Q over a countable domain X, we define the RD (of order 2) by $R(P\|Q) = \sum_{x\in X} P(x)^2/Q(x)$, with the convention that the fraction is zero when both the numerator and denominator are zero. We will use the following property: if P (resp. Q) is a direct product of independent distributions P_1 and P_2 (resp. Q_1 and Q_2), then we have $RD(P\|Q) = RD(P_1 \times P_2\|Q_1 \times Q_2) = RD(P_1\|P_2) \cdot RD(Q_1\|Q_2)$.

Lemma 10 (Lemma 2.9 in [6]). *Let P and Q denote distributions with* $\mathrm{Supp}(P) \subseteq \mathrm{Supp}(Q)$ *and $A \subseteq \mathrm{Supp}(Q)$ be arbitrary set. Then, we have $Q(A) \geq P(A)^2/R(P\|Q)$ where $P(A)$ and $Q(A)$ are measure of A under the distribution P and Q respectively.*

We also recall that the RD between two offset discrete Gaussians is bounded as follows.

Lemma 11 (Lemma 4.2 in [28]). *For any n-dimensional lattice $L \subseteq \mathbb{R}^n$ and invertible matrix $\mathbf{\Sigma}$, set $P = D_{\Lambda,\mathbf{\Sigma},\mathbf{w}}$ and $Q = D_{\Lambda,\mathbf{\Sigma},\mathbf{z}}$ for some fixed $\mathbf{w}, \mathbf{z} \in \Lambda$. Then, $R(P\|Q) \leq \exp(2\pi\|\mathbf{w} - \mathbf{z}\|^2/s_n(\mathbf{\Sigma})^2)$.*

Then, we proceed to the proof of Lemma 9.

Proof (of Lemma 9). Let $A \subseteq \mathbb{Z}^{n\times m}$ be the set of $\mathbf{X} = (\mathbf{X}_1|\mathbf{X}_2)$ such that $\mathbf{U}$ satisfying the properties listed in the statement does not exist. Theorem 8 implies that when $\mathbf{X}$ is sampled from the distribution $Q := D_{\mathbb{Z},\sigma_1}^{m_1\times m_2} \times (D_{\mathbb{Z}^{m_2},\sigma_2,\boldsymbol{\delta}_1} \times \cdots \times D_{\mathbb{Z}^{m_2},\sigma_2,\boldsymbol{\delta}_n})^t$, we have $Q(A) \leq 2^{-\Omega(n')}$. We want to prove that $P(A) = 2^{-\Omega(n')}$ for the distribution $P := D_{\mathbb{Z},\sigma_1}^{n\times m_1} \times D_{\mathbb{Z},\sigma_2}^{n\times m_2}$. By Lemma 10, we have $P(A) \leq \sqrt{Q(A) \cdot R(P\|Q)} \leq \sqrt{R(P\|Q)} \cdot 2^{-\Omega(n')}$. To complete the proof, it suffices to show $R(P\|Q) = O(1)$. We have

$$\begin{aligned} R(P\|Q) &= R\big(D_{\mathbb{Z},\sigma_1}^{n\times m_1} \times D_{\mathbb{Z},\sigma_2}^{n\times m_2} \| D_{\mathbb{Z},\sigma_1}^{m_1\times m_2} \times (D_{\mathbb{Z}^{m_2},\sigma_2,\boldsymbol{\delta}_1} \times \cdots \times D_{\mathbb{Z}^{m_2},\sigma_2,\boldsymbol{\delta}_n})\big) \\ &= R((D_{\mathbb{Z},\sigma_2})^n \| (D_{\mathbb{Z},\sigma_2,1})^n) \\ &\leq \exp(2\pi n/\sigma_2^2), \end{aligned}$$

where we use Lemma 11 in the last inequality. Since $\sigma_2 \geq \Omega(n^{1/2})$, we conclude that $R(P\|Q) = O(1)$. This completes the proof of Lemma 9. □

Next, we define the learning with errors (LWE) assumption. It was shown that the assumption holds as long as certain lattice problems are hard in the worst case [38, 36, 12].

Definition 12. For an integers $n = n(\lambda)$, $m = m(\lambda)$, $q = q(\lambda)$, a real number $\alpha(\lambda) \in (0,1)$, and an algorithm $\mathcal{A}$, the advantage for the learning with errors problem $\mathrm{LWE}_{n,m,q,\alpha}$ of $\mathcal{A}$ is defined as follows:

$$\big|\Pr[\mathcal{A}(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{x}) \to 1] - \Pr[\mathcal{A}(\mathbf{A}, \mathbf{w} + \mathbf{x}) \to 1]\big|$$

where $\mathbf{A} \hookleftarrow \mathbb{Z}_q^{n\times m}$, $\mathbf{s} \hookleftarrow \mathbb{Z}_q^n$, $\mathbf{w} \hookleftarrow \mathbb{Z}_q^m$, and $\mathbf{x} \hookleftarrow D_{\mathbb{Z},\alpha q}^m$. We say that $\mathrm{LWE}_{n,m,q,\alpha}$ assumption holds if the advantage is negligible for every probabilistic polynomial-time $\mathcal{A}$.

3 Trace and Revoke from Inner-Product Functional Encryption

In this section, we provide a generic transformation from a bounded collusion, random keys inner-product functional encryption scheme $\mathcal{FE}$ to a trace-and-revoke scheme $\mathcal{TR}$. Since intuition was provided in Section 1, we proceed directly to the formal construction. We remark that the informal description provided in Section 1 only described the scheme in the context of public traceability, but it can be readily adapted to handle almost public traceability.

3.1 The Scheme

We construct a trace-and-revoke scheme $\mathcal{TR}$ following the specifications of Definition 1. Our scheme assumes the existence of a public directory pd which contains the identities of the users that have been assigned keys in the system. The public directory is initially empty. We assume that pd can only be modified by a central authority (the key generator).

1. $\mathsf{Setup}(1^\lambda, 1^t, 1^r)$. Upon input the security parameter λ, the bound t on the number of traitors and the bound r on the number of revoked users, proceed as follows:
   (a) Let $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathcal{FE}.\mathsf{Setup}(1^\lambda, 1^\ell)$, where $\ell = t + r + 1$.
   (b) Output the public key pk and master secret key msk.
2. $\mathsf{KeyGen}(\mathsf{msk}, \mathsf{id})$. Upon input the master secret key msk and a user identity $\mathsf{id} \in \mathsf{ID}$, proceed as follows:
   (a) Sample $\mathsf{x}_{\mathsf{id}} \hookleftarrow \mathbb{Z}_p^\ell$. The pair $p_{\mathsf{id}} = (\mathsf{id}, \mathsf{x}_{\mathsf{id}})$ is appended to the public directory pd.
   (b) Let $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathcal{FE}.\mathsf{KeyGen}(\mathsf{msk}, \mathsf{x}_{\mathsf{id}})$.

(c) Output $\mathsf{sk}_{\mathsf{id}}$.

3. $\mathsf{Enc}(\mathsf{pd}, \mathsf{pk}, \mathcal{R}, m)$. Upon input the public key pk, a set of revoked users $\mathcal{R}$ of cardinality $\leq r$ and a plaintext messages $m \in \mathcal{M} = \mathbb{Z}_p$, proceed as follows:
   (a) Compute $\mathbf{v}_{\mathcal{R}} \in \mathbb{Z}_p^\ell \setminus \{\mathbf{0}\}$ such that $\langle \mathbf{x}_{\mathsf{id}}, \mathbf{v}_{\mathcal{R}} \rangle = 0$ for every $\mathsf{id} \in \mathcal{R}$.
   (b) Compute $\mathbf{y}_{\mathcal{R}} = m \cdot \mathbf{v}_{\mathcal{R}}$.
   (c) Output $C = (C_1, C_2) = (\mathcal{FE}.\mathsf{Enc}(\mathsf{pk}, \mathbf{y}_{\mathcal{R}}), \mathcal{R})$.
4. $\mathsf{Dec}(\mathsf{pd}, \mathsf{sk}_{\mathsf{id}}, C)$. Upon input the secret key $\mathsf{sk}_{\mathsf{id}}$ for user id and a ciphertext $C = (C_1, C_2)$, proceed as follows:
   (a) Parse C_2 as $C_2 = \mathcal{R}$. If $\mathsf{id} \in \mathcal{R}$, then abort.
   (b) Compute $\mathbf{v}_{\mathcal{R}} \in \mathbb{Z}_p^\ell \setminus \{\mathbf{0}\}$ such that $\langle \mathbf{x}_{\mathsf{id}}, \mathbf{v}_{\mathcal{R}} \rangle = 0$ for every $\mathsf{id} \in \mathcal{R}$.
   (c) Compute and output $m = \mathcal{FE}.\mathsf{Dec}(\mathsf{sk}_{\mathsf{id}}, C_1) / \langle \mathbf{x}_{\mathsf{id}}, \mathbf{v}_{\mathcal{R}} \rangle$.
5. $\mathsf{Trace}(\mathsf{pd}, \mathcal{R}, \mathcal{S}, \mathcal{O}^{\mathcal{D}})$. Upon input the public directory pd, a revoked set of users $\mathcal{R}$, a suspect set $\mathcal{S}$ of users and given access to the oracle $\mathcal{O}^{\mathcal{D}}$, first proceed as follows:
   (a) Find $m, m' \in \mathcal{M}$ such that the following quantity is non-negligible:
   $$\left| \Pr_{C \hookleftarrow \mathsf{Enc}(\mathsf{pd},\mathsf{pk},\mathcal{R},m)} \left[\mathcal{O}^{\mathcal{D}}(C, m) = 1 \right] - \Pr_{C' \hookleftarrow \mathsf{Enc}(\mathsf{pd},\mathsf{pk},\mathcal{R},m')} \left[\mathcal{O}^{\mathcal{D}}(C', m) = 1 \right] \right|.$$
   (b) Set $\mathcal{S}_1 = \{\mathsf{id}_1, \mathsf{id}_2, \ldots\} = \mathcal{S} \setminus \mathcal{R}$.
   (c) Compute $\mathbf{v}_{\mathcal{R}} \in \mathbb{Z}_p^\ell \setminus \{\mathbf{0}\}$ such that $\langle \mathbf{x}_{\mathsf{id}}, \mathbf{v}_{\mathcal{R}} \rangle = 0$ for every $\mathsf{id} \in \mathcal{R}$.
   Then execute the following steps with $i = 1, 2, \ldots$:
   (d) If $i = 1$, set $\mathbf{v}_{\mathcal{S}_i} = \mathbf{0}$. If $\mathcal{S}_i = \emptyset$, set $\mathbf{v}_{\mathcal{S}_i} = (m' - m) \cdot \mathbf{v}_{\mathcal{R}}$. Else compute $\mathbf{v}_{\mathcal{S}_i} \in \mathbb{Z}_p^\ell$ such that:
      i. $\langle \mathbf{x}_{\mathsf{id}}, \mathbf{v}_{\mathcal{S}_i} \rangle = 0$ for every $\mathsf{id} \in \mathcal{S}_i \cup \mathcal{R}$.
      ii. $\langle \mathbf{x}_{\mathsf{id}}, \mathbf{v}_{\mathcal{S}_i} \rangle = (m' - m) \cdot \langle \mathbf{x}_{\mathsf{id}}, \mathbf{v}_{\mathcal{R}} \rangle$ for every $\mathsf{id} \in \mathcal{S}_1 \setminus \mathcal{S}_i$.
   (e) Repeat the following steps sufficiently many times (as dictated by Hoeffding's inequality) to compute an approximation of the probability p_i that the response from $\mathcal{O}^{\mathcal{D}}$ is $b_i = 1$.
      i. Construct $\mathbf{y} = \mathbf{v}_{\mathcal{S}_i} + m \cdot \mathbf{v}_{\mathcal{R}} \in \mathbb{Z}_p^\ell$;
      ii. The probe ciphertext is $C^{\mathcal{S}_i} = (\mathcal{FE}.\mathsf{Enc}(\mathsf{pk}, \mathbf{y}), \mathcal{R})$;
      iii. Provide the oracle $\mathcal{O}^{\mathcal{D}}$ with $(C^{\mathcal{S}_i}, m)$ as input and get a binary value b_i as output.
   (f) If $i > 1$ and $|p_i - p_{i-1}|$ is non-negligible, then output id_{i-1} and abort;
   (g) If $\mathcal{S}_i = \emptyset$, then output $\perp$ and abort; else, set $\mathcal{S}_{i+1} = \mathcal{S}_i \setminus \{\mathsf{id}_i\}$.

For the tracing security proof, we require that in Step (a) of Algorithm Enc, in Step (b) of Algorithm Dec and in Step (c) of Algorithm Trace, the vector $\mathbf{v}_{\mathcal{R}}$ be uniquely determined by $\mathcal{R}$, in the same unique way across all algorithms. One way of achieving this property is to order the $\mathbf{x}_{\mathsf{id}}$'s for $\mathsf{id} \in \mathcal{R}$ lexicographically, and run a deterministic linear system solver. We proceed in the same way (using always the same deterministic algorithm) for vector $\mathbf{v}_{\mathcal{S}_i}$ at Step (d) of Algorithm Trace.

The tuples in pd are used for full public traceability where an untrusted machine can trace the id of a traitor. Alternatively, to enable almost public traceability, the $\mathbf{x}_{\mathsf{id}}$'s only may be stored in pd. In that case, a trusted server has to be used to store the pairs $(\mathsf{id}, \mathbf{x}_{\mathsf{id}})$. Further, the set $\mathcal{R}$ of revoked users may be replaced by the set of $\mathbf{x}_{\mathsf{id}}$'s for $\mathsf{id} \in \mathcal{R}$.

We first check the correctness of the scheme.

Theorem 13. *Assume that $p = \lambda^{\omega(1)}$. Let $\mathcal{R}$ be a set of revoked users of cardinality $\leq r$. Then, for every $\mathsf{id} \notin \mathcal{R}$ and every $m \in \mathcal{M} = \mathbb{Z}_p$, we have*

$$\mathsf{Dec}(\mathsf{pd}, \mathsf{sk}_{\mathsf{id}}, \mathsf{Enc}(\mathsf{pd}, \mathsf{pk}, \mathcal{R}, m)) = m,$$

with probability $\geq 1 - \lambda^{-\omega(1)}$.

Proof. As $\mathbf{x}_{\mathsf{id}}$ is uniform in $\mathbb{Z}_p^\ell$, and thanks to the parameter choices of $p = \lambda^{\omega(1)}$ and $\ell > r$, we have that $\langle \mathbf{x}_{\mathsf{id}}, \mathbf{v}_{\mathcal{R}} \rangle \neq 0$, with overwhelming probability. The execution of $\mathsf{Dec}(\mathsf{pd}, \mathsf{sk}_{\mathsf{id}}, C)$, with $C = (C_1, C_2) = \mathsf{Enc}(\mathsf{pd}, \mathsf{pk}, \mathcal{R}, m)$, proceeds to Step (b) and computes (with overwhelming probability):

$$\mathsf{Dec}(\mathsf{pd}, \mathsf{sk}_{\mathsf{id}}, C) = \frac{\mathcal{FE}.\mathsf{Dec}(\mathsf{sk}_{\mathsf{id}}, C_1)}{\langle \mathbf{x}_{\mathsf{id}}, \mathbf{v}_{\mathcal{R}} \rangle} = \frac{\langle \mathbf{x}_{\mathsf{id}}, m \cdot \mathbf{v}_{\mathcal{R}} \rangle}{\langle \mathbf{x}_{\mathsf{id}}, \mathbf{v}_{\mathcal{R}} \rangle} = m,$$

by correctness of $\mathcal{FE}$. □

Now, we consider the implementation of Step (a) of Algorithm Trace. The aim is to find $m, m' \in \mathbb{Z}_p$ such that an encryption of m has a non-negligible probability difference of decrypting to m and m' via $\mathcal{O}^{\mathcal{D}}$. These plaintexts are used for tracing as follows: the first probe ciphertext distribution will be a genuine encryption of m, while the last probe ciphertext distribution will be a genuine encryption of m'. (To see this, observe that for the last probe ciphertext, we have $\mathcal{S}_i = \emptyset$ and $\mathbf{v}_{\mathcal{S}_i} = (m' - m) \cdot \mathbf{v}_{\mathcal{R}}$. Consequently, we have $C^{\mathcal{S}_i} = (\mathcal{FE}.\mathsf{Enc}(\mathsf{pk}, \mathbf{y}), \mathcal{R})$ where $\mathbf{y} = \mathbf{v}_{\mathcal{S}_i} + m \cdot \mathbf{v}_{\mathcal{R}} = m' \cdot \mathbf{v}_{\mathcal{R}}$.) The fact that $\mathcal{O}^{\mathcal{D}}$ behaves differently for these two distributions ensures that there will be an i such that $|p_i - p_{i-1}|$ is non-negligible. Now, if the oracle $\mathcal{O}^{\mathcal{D}}$ was perfect, i.e., a genuine encryption of m always decrypts to m for all m, then the existence of a pair (m, m') as in Step (a) would be immediate. The difficulty is that the oracle $\mathcal{O}^{\mathcal{D}}$ only achieves correct decryption with non-negligible advantage.

Lemma 14. *Let $\mathcal{R}$ be arbitrary and assume that Equation* (1) *holds for $\mathcal{R}$. Then, with probability $\geq 1/(4\lambda^c)$ over the choice of $m, m' \hookleftarrow \mathcal{M}$, we have:*

$$\Big| \Pr_{C \hookleftarrow \mathsf{Enc}(\mathsf{pk}, \mathcal{R}, m)} \big[\mathcal{O}^{\mathcal{D}}(C, m) = 1\big] - \Pr_{C' \hookleftarrow \mathsf{Enc}(\mathsf{pk}, \mathcal{R}, m')} \big[\mathcal{O}^{\mathcal{D}}(C', m) = 1\big] \Big| \geq \frac{1}{2\lambda^c}.$$

Based on Lemma 14, Step (a) of Algorithm Trace can be implemented by repeatedly sampling $m, m' \hookleftarrow \mathcal{M}$ and estimating the probabilities that $\mathcal{O}^{\mathcal{D}}(C, m) = 1$ and $\mathcal{O}^{\mathcal{D}}(C', m) = 1$ using Hoeffding's bound, until the probability difference is sufficiently large.

Proof. For $m, m' \in \mathcal{M}$, let $P(m', m)$ denote the probability that $\mathcal{O}^{\mathcal{D}}(C', m) = 1$, where $C' \hookleftarrow \mathsf{Enc}(\mathsf{pd}, \mathsf{pk}, \mathcal{R}, m')$. Equation (1) states that

$$\Pr_{m \hookleftarrow \mathcal{M}} [P(m, m)] \geq \frac{1}{|\mathcal{M}|} + \frac{1}{\lambda^c}.$$

Let us assume by contradiction (of the statement to be proved), that

$$\Pr_{m, m' \hookleftarrow \mathcal{M}} [|P(m, m) - P(m', m)| < \frac{1}{2\lambda^c}] > 1 - \frac{1}{4\lambda^c}. \tag{2}$$

We show that if (2) holds, then the following inequality holds as well.

$$\Pr_{m' \hookleftarrow \mathcal{M}} [\Pr_{m \hookleftarrow \mathcal{M}} [|P(m, m) - P(m', m)| < \frac{1}{2\lambda^c}] > 1 - \frac{1}{2\lambda^c}] > \frac{1}{2}. \tag{3}$$

By contradiction of (3) above, let us assume that

$$\Pr_{m' \hookleftarrow \mathcal{M}} [\Pr_{m \hookleftarrow \mathcal{M}} [|P(m, m) - P(m', m)| < \frac{1}{2\lambda^c}] > 1 - \frac{1}{2\lambda^c}] \leq \frac{1}{2}.$$

We consider two types of m', depending whether $\Pr_m[|P(m,m) - P(m',m)| < \frac{1}{2\lambda^c}]$ is greater than $1 - \frac{1}{2\lambda^c}$ (Type 1) or not (Type 2). Let $x \leq 1/2$ be the proportion of m''s of the first type. Then we would have

$$\begin{aligned}
\Pr_{m,m'}[|P(m,m) - P(m',m)| < \frac{1}{2\lambda^c}] &= \Pr_{m'}[\Pr_{m}[|P(m,m) - P(m',m)| < \frac{1}{2\lambda^c}]] \\
&= \frac{1}{|\mathcal{M}|} \sum_{\substack{m' \\ \text{of Type 1}}} \Pr_m [|P(m,m) - P(m',m)| < \frac{1}{2\lambda^c}] \\
&+ \frac{1}{|\mathcal{M}|} \sum_{\substack{m' \\ \text{of Type 2}}} \Pr_m [|P(m,m) - P(m',m)| < \frac{1}{2\lambda^c}] \\
&\leq \frac{1}{|\mathcal{M}|} \sum_{\substack{m' \\ \text{of Type 1}}} 1 + \frac{1}{|\mathcal{M}|} \sum_{\substack{m' \\ \text{of Type 2}}} (1 - \frac{1}{2\lambda^c}) \\
&= x + (1-x)(1 - \frac{1}{2\lambda^c}) \leq 1 - \frac{1}{4\lambda^c},
\end{aligned}$$

which would contradict (2) above.

We consider an m' of Type 1. Using the fact that $\sum_m P(m',m) \leq 1$, we obtain:

$$\sum_m P(m,m) < \frac{|\mathcal{M}|}{2\lambda^c} + \sum_m \left(P(m',m) + \frac{1}{2\lambda^c}\right) \leq 1 + \frac{|\mathcal{M}|}{\lambda^c}.$$

This contradicts Equation (1). □

3.2 Semantic Security

We start by proving IND-CPA security of our scheme.

Theorem 15. *If $\mathcal{FE}$ is r-*IND-CPA *secure, then $\mathcal{TR}$ is* IND-CPA *secure.*

Proof. Let $\mathcal{A}_{\mathcal{TR}}$ be a probabilistic polynomial-time adversary that breaks semantic security of $\mathcal{TR}$. We construct a probabilistic polynomial-time adversary $\mathcal{A}_{\mathcal{FE}}$ that breaks semantic security of $\mathcal{FE}$. Adversary $\mathcal{A}_{\mathcal{FE}}$ proceeds as follows.

- It first obtains the public key pk output by the $\mathcal{FE}$ challenger (who runs the $\mathcal{FE}.\mathsf{Setup}(1^\lambda, 1^\ell)$ algorithm) and relays it to $\mathcal{A}_{\mathcal{TR}}$.
- The adversary $\mathcal{A}_{\mathcal{TR}}$ adaptively chooses at most r identities id (that forms the revoked set $\mathcal{R}$) and are included in pd. The adversary $\mathcal{A}_{\mathcal{FE}}$ then queries the $\mathcal{FE}$ challenger for each $\mathbf{x}_{\mathsf{id}}$ for all $\mathsf{id} \in \mathcal{R}$ and receives the corresponding $\mathsf{sk}_{\mathsf{id}}$. Adversary $\mathcal{A}_{\mathcal{FE}}$ relays all $\mathsf{sk}_{\mathsf{id}}$ for each $\mathsf{id} \in \mathcal{R}$ to $\mathcal{A}_{\mathcal{TR}}$.
- When $\mathcal{A}_{\mathcal{TR}}$ chooses two messages $m_0, m_1 \in \mathcal{M}$ and provides them to $\mathcal{A}_{\mathcal{FE}}$, adversary $\mathcal{A}_{\mathcal{FE}}$ proceeds as follows:
  - It computes $\mathbf{v}_{\mathcal{R}} \in \mathbb{Z}_p^\ell \setminus \{\mathbf{0}\}$ such that $\langle \mathbf{x}_{\mathsf{id}}, \mathbf{v}_{\mathcal{R}} \rangle = 0$ for every $\mathsf{id} \in \mathcal{R}$.
  - It sends $\mathbf{y}_{\mathcal{R},0} = m_0 \cdot \mathbf{v}_{\mathcal{R}}$ and $\mathbf{y}_{\mathcal{R},1} = m_1 \cdot \mathbf{v}_{\mathcal{R}}$ to the $\mathcal{FE}$ challenger who samples $b \hookleftarrow \{0,1\}$ and encrypts $\mathbf{y}_{\mathcal{R},b}$ as $C_{\mathbf{y}_{\mathcal{R},b}} \hookleftarrow \mathcal{FE}.\mathsf{Enc}(\mathsf{pk}, \mathbf{y}_{\mathcal{R},b})$.
  - Adversary $\mathcal{A}_{\mathcal{FE}}$ receives $C_{\mathbf{y}_{\mathcal{R},b}}$ from the $\mathcal{FE}$ challenger and sends $C = (C_{\mathbf{y}_{\mathcal{R},b}}, \mathcal{R})$ to $\mathcal{A}_{\mathcal{TR}}$.
- Finally, adversary $\mathcal{A}_{\mathcal{TR}}$ outputs its guess $b' \in \{0,1\}$ and $\mathcal{A}_{\mathcal{FE}}$ also outputs b' as its own guess of b.

Note that adversary $\mathcal{A}_{\mathcal{FE}}$ behaves as an $\mathsf{IND\text{-}CPA}$ challenger in the view of $\mathcal{A}_{\mathcal{TR}}$. Further, it is a valid adversary against $\mathcal{FE}$, as $\langle \mathbf{y}_{\mathcal{R},0}, \mathbf{x}_{\mathsf{id}} \rangle = \langle \mathbf{y}_{\mathcal{R},1}, \mathbf{x}_{\mathsf{id}} \rangle$ for every vector $\mathbf{x}_{\mathsf{id}}$ queried to the $\mathcal{FE}$ challenger (i.e., each $\mathsf{id} \in \mathcal{R}$). The advantage of $\mathcal{A}_{\mathcal{FE}}$ is exactly the same as the advantage of $\mathcal{A}_{\mathcal{TR}}$. □

We may observe that for $\mathcal{TR}$ to be $\mathsf{IND\text{-}CPA}$ secure, an r-$\mathsf{IND\text{-}CPA}$ secure $\mathcal{FE}$ scheme is sufficient. However, as we see below, for traceability with up to t colluding traitors along with r already revoked users, we need an $\mathcal{FE}$ scheme that is $(t+r)$-$\mathsf{IND\text{-}CPA}$ secure.

3.3 Traceability

We assume that the decoder $\mathcal{D}$ produced by the adversary $\mathcal{A}$ is useful as indicated in Equation (1). The Trace algorithm has access to this decoder through the oracle $\mathcal{O}^{\mathcal{D}}$. We also assume that $\mathcal{T} \subseteq \mathcal{S}$, as specified in game $\mathsf{AD\text{-}TT}$. Without loss of generality, we may assume $|\mathcal{S}| = t$. We show in this section that $\mathsf{Trace}(\mathsf{pd}, \mathcal{R}, \mathcal{S}, \mathcal{O}^{\mathcal{D}})$ outputs an identity $\mathsf{id} \in \mathcal{T}$.

Theorem 16. *If $\mathcal{FE}$ is $(t+r)$-$\mathsf{IND\text{-}CPA}$ secure, then $\mathcal{TR}$ satisfies public traceability.*

Proof. Let us assume by contradiction that an adversary $\mathcal{A}$ can break the public traceability of $\mathcal{TR}$ with non-negligible probability. We then construct a probabilistic polynomial-time adversary $\mathcal{A}_{\mathcal{FE}}$ that breaks the semantic security of $\mathcal{FE}$. Adversary $\mathcal{A}_{\mathcal{FE}}$ proceeds as follows.

- It first obtains the public key pk output by the $\mathcal{FE}$ challenger (who runs the $\mathcal{FE}.\mathsf{Setup}(1^\lambda, 1^\ell)$ algorithm) and relays it to the adversary $\mathcal{A}$.
- When $\mathcal{A}$ asks $\mathcal{A}_{\mathcal{FE}}$ to create a p_{id} for some id, adversary $\mathcal{A}_{\mathcal{FE}}$ in turn asks the $\mathcal{FE}$ challenger to do the same. The $\mathcal{FE}$ challenger randomly chooses a vector $\mathbf{x}_{\mathsf{id}} \hookleftarrow \mathbb{Z}_p^\ell$ and sends it to $\mathcal{A}_{\mathcal{FE}}$ who further relays it to $\mathcal{A}$.
- When $\mathcal{A}$ makes a key query for an identity id, adversary $\mathcal{A}_{\mathcal{FE}}$ queries the $\mathcal{FE}$ challenger for a secret key. Adversary $\mathcal{A}_{\mathcal{FE}}$ receives the corresponding $\mathsf{sk}_{\mathsf{id}}$ from the $\mathcal{FE}$ challenger and relays it to $\mathcal{A}$.
- When $\mathcal{A}$ chooses a set $\mathcal{R}$ of up to r revoked users, adversary $\mathcal{A}_{\mathcal{FE}}$ makes $|\mathcal{R}|$ key queries to the $\mathcal{FE}$ challenger. Adversary $\mathcal{A}_{\mathcal{FE}}$ is given the set $\mathsf{sk}_{\mathsf{id}}$'s of corresponding secret keys that is relayed to $\mathcal{A}$.

Note that since $\mathcal{A}$ makes at most t key queries and $|\mathcal{R}| \leq r$, adversary $\mathcal{A}_{\mathcal{FE}}$ makes at most $t+r$ key queries for the $\mathcal{FE}$ challenger.

- Adversary $\mathcal{A}$ finally produces a pirate decoder $\mathcal{D}$ [4] and chooses a suspect set $\mathcal{S}$ of cardinality $\leq t$ that contains $\mathcal{T}$. Then, the adversary $\mathcal{A}_{\mathcal{FE}}$ executes the Trace algorithm on $\mathcal{O}^{\mathcal{D}}$ to find i such that $|p_i - p_{i-1}|$ is non-negligible. If Trace outputs $\perp$ or index i such that $\mathsf{id}_{i-1} \notin \mathcal{T}$ but $\mathsf{id}_{i-1} \in \mathcal{S}$, then $\mathcal{A}_{\mathcal{FE}}$ outputs a random bit. Otherwise, it sets $\mathbf{y}_0 = \mathbf{v}_{\mathcal{S}_{i-1}} + m \cdot \mathbf{v}_{\mathcal{R}}$ and $\mathbf{y}_1 = \mathbf{v}_{\mathcal{S}_i} + m \cdot \mathbf{v}_{\mathcal{R}}$, and sends them as challenge messages to the $\mathcal{FE}$ challenger.[5]
- The $\mathcal{FE}$ challenger samples $b \hookleftarrow \{0,1\}$ and sends $\mathcal{FE}.\mathsf{Enc}(\mathsf{pk}, \mathbf{y}_b)$ to $\mathcal{A}_{\mathcal{FE}}$. The adversary $\mathcal{A}_{\mathcal{FE}}$ runs $\mathcal{O}^{\mathcal{D}}$ on input (C_b, m), where $C_b = (\mathcal{FE}.\mathsf{Enc}(\mathsf{pk}, \mathbf{y}_b), \mathcal{R})$. Then $\mathcal{O}^{\mathcal{D}}$ outputs the bit $b' \in \{0,1\}$.

[4] Recall that we assume that $\mathcal{D}$ is stateless/resettable and replies independently to successive queries.

[5] Here, m and m' are chosen as in Step (a), $\mathbf{v}_{\mathcal{R}} \in \mathbb{Z}_p^\ell$ is chosen as in Step (c), and $\mathbf{v}_{\mathcal{S}_{i-1}}, \mathbf{v}_{\mathcal{S}_i} \in \mathbb{Z}_p^\ell$ are chosen as in Step (d) of algorithm Trace.

- Finally, adversary $\mathcal{A}_{\mathcal{FE}}$ outputs the same bit $b' \in \{0,1\}$ if $p_i - p_{i-1} > 0$ and $1-b'$ otherwise.

We first argue that $\mathcal{A}_{\mathcal{FE}}$ is a valid adversary against the $\mathcal{FE}$ challenger. Recall that when Trace does not abort in Step (f) for a value of i, we have $\mathsf{id}_{i-1} \notin \mathcal{T}$ but $\mathsf{id}_{i-1} \in \mathcal{S}$. The keys queried by $\mathcal{A}_{\mathcal{FE}}$ are for $\mathsf{id} \in \mathcal{R} \cup \mathcal{T}$. This set $\mathcal{R} \cup \mathcal{T}$ can be partitioned into $\mathcal{R} \cup (\mathcal{T} \cap \mathcal{S}_{i-1})$ and $\mathcal{T} \cap (\mathcal{S}_1 \setminus \mathcal{S}_{i-1})$.

1. For $\mathsf{id} \in \mathcal{R}$, we have $\langle \mathbf{x}_{\mathsf{id}}, \mathbf{y}_0 \rangle = \langle \mathbf{x}_{\mathsf{id}}, \mathbf{y}_1 \rangle = 0$. For $\mathsf{id} \in \mathcal{S}_{i-1}$, we have $\langle \mathbf{x}_{\mathsf{id}}, \mathbf{y}_0 \rangle = \langle \mathbf{x}_{\mathsf{id}}, \mathbf{y}_1 \rangle = m \cdot \langle \mathbf{x}_{\mathsf{id}}, \mathbf{v}_{\mathcal{R}} \rangle$. Hence for all $\mathsf{id} \in \mathcal{R} \cup (\mathcal{T} \cap \mathcal{S}_{i-1})$ for which the $\mathsf{sk}_{\mathsf{id}}$ was queried by $\mathcal{A}_{\mathcal{FE}}$, the inner products $\langle \mathbf{x}_{\mathsf{id}}, \mathbf{y}_0 \rangle$ and $\langle \mathbf{x}_{\mathsf{id}}, \mathbf{y}_1 \rangle$ have the same value.
2. Similarly, for $\mathsf{id} \in \mathcal{T} \cap (\mathcal{S}_1 \setminus \mathcal{S}_{i-1})$, we have $\langle \mathbf{x}_{\mathsf{id}}, \mathbf{y}_0 \rangle = \langle \mathbf{x}_{\mathsf{id}}, \mathbf{y}_1 \rangle = m' \cdot \langle \mathbf{x}_{\mathsf{id}}, \mathbf{v}_{\mathcal{R}} \rangle$.

Hence, $\mathcal{A}_{\mathcal{FE}}$ is a valid adversary against the $\mathcal{FE}$ challenger.

We recollect that in the $\mathsf{AD\text{-}TT}$ game, we say that $\mathcal{A}$ wins if the decryption box $\mathcal{D}$ output by it is such that when Trace is executed on input $\mathcal{O}^{\mathcal{D}}$, it fails to identify a traitor. In such a case, Trace either outputs $\perp$ or it outputs an $\mathsf{id}_{i-1} \notin \mathcal{T}$ with probability at least $1/\lambda^c$. We next argue that if $\mathcal{A}$ outputs $\mathcal{D}$ that satisfies this winning condition of the $\mathsf{AD\text{-}TT}$ game, then $\mathcal{A}_{\mathcal{FE}}$ has non-negligible advantage in the above game. To see this, we first observe that when $\mathcal{A}_{\mathcal{FE}}$ returns a random bit, it correctly guesses b with probability $1/2$. Then, it suffices to show the following:

- In the above game, adversary $\mathcal{A}_{\mathcal{FE}}$ does not return a random bit with non-negligible probability.
- Conditioned on $\mathcal{A}_{\mathcal{FE}}$ not returning a random bit, Trace outputs id_{i-1} such that $|p_i - p_{i-1}|$ is non-negligible.

The second item follows because if $|p_i - p_{i-1}|$ is not sufficiently large, Trace does not output id_{i-1} at Step (f) of Trace except for a negligible probability (because of the Hoeffding bound). Next, we prove the first item. Since we are assuming $\mathcal{D}$ satisfies the winning condition, when Trace is executed on input $\mathcal{O}^{\mathcal{D}}$, it outputs $\perp$ or it outputs an $\mathsf{id}_{i-1} \notin \mathcal{T}$ with probability at least $1/\lambda^c$. On the other hand, we observe that $\mathcal{A}_{\mathcal{FE}}$ does not return a random bit only when Trace on input $\mathcal{O}^{\mathcal{D}}$ outputs $\mathsf{id}_{i-1} \notin \mathcal{T}$. Therefore, to conclude the proof, it suffices to show the following lemma, which says that the probability of Trace returning $\perp$ is negligible. □

Lemma 17. *Assume that a pirate decoder $\mathcal{D}$ satisfies Equation* (1)*. Then, the execution of* Trace *does not return $\perp$ but returns some* $\mathsf{id} \in \mathcal{S}$ *with overwhelming probability.*

Proof. We consider a variant of Trace that continues its execution until it exhausts $\mathcal{S} \setminus \mathcal{R}$, even if it has already output an id. We consider the probabilities p_i at the start and end of that modified execution.

1. At the beginning, algorithm Trace considers $\mathcal{S}_1 = \mathcal{S} \setminus \mathcal{R}$ and $\mathbf{v}_{\mathcal{S}_1} = \mathbf{0}$. Hence, the genuine ciphertext output by the Enc algorithm and the probe ciphertext created by the Trace algorithm for the suspect subset $\mathcal{S}_1$ are exactly the same.
2. When $i = |\mathcal{S} \setminus \mathcal{R}| + 1$, we have $\mathcal{S}_i = \emptyset$ and $\mathbf{v}_{\mathcal{S}_i} = (m' - m) \cdot \mathbf{v}_{\mathcal{R}}$. In Step (a) of the Trace algorithm, the messages m and m' were chosen such that the difference in the probabilities p_1 and $p_{|\mathcal{S} \setminus \mathcal{R}|+1}$ is $\geq 1/(2\lambda^c)$.

Note that the two latter observations imply, via the triangle inequality, that there exists an i such that $|p_i - p_{i-1}|$ is non-negligible. By the Hoeffding bound, Trace algorithm outputs id_{i-1} with overwhelming probability. □

4 Trace and Revoke from Learning with Errors

Recall that Agrawal et al. [4] provided a construction for inner product functional encryption from LWE. Instantiating our generic transformation of Section 3 with this scheme is possible, but leads to reliance on LWE with subexponential error rates. In Subsection 4.2, we provide a new construction of an inner product functional encryption scheme from LWE in a much weaker model than that considered in [4]. We restrict to the setting of bounded collusions and also crucially exploit the fact that the adversary's key requests are random vectors for our application as described in Section 3. The performances of both resulting trace-and-revoke systems are discussed in Subsection 4.1.

4.1 Two Trace-and-Revoke Constructions

Our IPFE to trace-and-revoke generic transformation cannot be directly instantiated with the LWE-based IPFE over $\mathbb{Z}_p$ from [4], because the key generation algorithm of the latter is stateful: it keeps track of all the secret keys it has generated. The statefulness necessity may be explained as follows. The master secret key is an *integer* matrix with small entries. When the attacker makes a key query for a vector *modulo p*, it learns the *integer* product between a conversion to the integers of that vector and the master secret key. If the key generation algorithm does not maintain a state, then it does not seem possible to prevent an adversary from making key queries for vectors that are linearly dependent *modulo p* but linearly independent over the *integers*: the attacker could then make valid key queries but still learn the master secret key.

The Key Generation State is Unnecessary. In [4], it was noted that if the vectors queried by the adversary are guaranteed to be linearly independent modulo p, then there is no need for a stateful key generation algorithm. In our case, there are as many vectors as users, each vector is uniformly sampled from $\mathbb{Z}_p^\ell$ and the adversary has access to $\leq r+t < \ell$ vectors. By setting $p = 2^{\Omega(\lambda)}$, the probability that there exists a subset of t key vectors that are linearly independent is $2^{-\Omega(\lambda)}$. We can then remove the state in the LWE-based IPFE over $\mathbb{Z}_p^\ell$, and apply the transformation from the previous section.

The resulting trace-and-revoke scheme inherits the unsatisfactory performance of its underlying IPFE (see [4, Section 4.2] for further details), stemming from the subexponential error rate in the LWE hardness assumption.

Large LWE Errors are Unnecessary. In Subsection 4.2 below, we exploit the randomness of the key queries further, as well as the bounded number of queries (as allowed by our trace-and-revoke application). We obtain a random-key bounded-collusion FE for inner products from LWE with significantly better parameters. In particular, we rely on slightly super-polynomial error rates for LWE, which allows to take smaller parameters.[6]

Both the public key and master secret key of the resulting trace-and-revoke scheme consist of $\widetilde{O}((t+r+\lambda)\lambda)$ bits. To every user id corresponds a secret key $\mathsf{sk}_{\mathsf{id}}$ of bit-length $\widetilde{O}(t+r+\lambda)$ and a vector p_{id} of bit-length $\widetilde{O}(t+r+\lambda)$. Algorithm Enc maps a plaintext in $\{0,1\}$ to a ciphertext of bit-length $\widetilde{O}(t+r+\lambda)$.

[6] We observe that the scheme from Subsection 4.2 allows for polynomial error rates, but the correctness of our trace-and-revoke construction requires $p \geq \lambda^{\omega(1)}$, which leads to a $\lambda^{\omega(1)}$ LWE error rate in our IPFE.

4.2 Bounded Collusion FE for Inner Products from LWE

The construction we provide here relies on LWE with a small error rate and hence small modulus and dimension.

Construction. Let p be the modulus of the scheme, 2ℓ be the dimension of the scheme, and L be the upper bound on the size of the collusion.

- $\mathcal{FE}.\mathsf{Setup}(1^\lambda, 1^L, 1^{2\ell})$. Set integers n, m, $q = p^k$ for some integer $k \geq 2$, and reals $\alpha \in (0,1)$ and $\sigma_0, \sigma_1, \sigma_2 > 0$, as explained below. Sample $\mathbf{A} \hookleftarrow \mathbb{Z}_q^{m\times n}$ and $\mathbf{Z} \hookleftarrow D_{\mathbb{Z},\sigma_0}^{2\ell\times m}$. Compute $\mathbf{U} = \mathbf{Z}\cdot\mathbf{A} \in \mathbb{Z}_q^{2\ell\times n}$. Define
$$\mathsf{msk} := \mathbf{Z} \text{ and } \mathsf{pk} := (\mathbf{A}, \mathbf{U}).$$
- $\mathcal{FE}.\mathsf{KeyGen}(\mathsf{msk}, \mathbf{x})$. Given $\mathbf{x} = (x_1, \ldots, x_{2\ell})^t \in \mathbb{Z}_p^{2\ell}$, sample $\bar{x}_i \hookleftarrow D_{p\mathbb{Z}+x_i,\sigma_1}$ for $i \in [\ell]$ and $\bar{x}_i \hookleftarrow D_{p\mathbb{Z}+x_i,\sigma_2}$ for $i \in [\ell+1, 2\ell]$. Set $\bar{\mathbf{x}} := (\bar{x}_1, \ldots, \bar{x}_{2\ell})^t \in \mathbb{Z}^{2\ell}$ and $\mathbf{z}_\mathbf{x}^t = \bar{\mathbf{x}}^t \cdot \mathbf{Z} \in \mathbb{Z}^m$. Note that we have $\bar{\mathbf{x}} \equiv \mathbf{x} \bmod p$ by construction. Finally, return $\mathsf{sk}_\mathbf{x} = (\bar{\mathbf{x}}, \mathbf{z}_\mathbf{x})$.
- $\mathcal{FE}.\mathsf{Enc}(\mathsf{pk}, \mathbf{y})$. To encrypt a vector $\mathbf{y} \in \mathbb{Z}_p^{2\ell}$, sample $\mathbf{s} \hookleftarrow \mathbb{Z}_q^n$, $\mathbf{e}_0, \mathbf{e}_1 \hookleftarrow D_{\mathbb{Z},\alpha q}^m$ and compute
$$\mathbf{c}_0 = \mathbf{A}\mathbf{s} + \mathbf{e}_0 \in \mathbb{Z}_q^m, \qquad \mathbf{c}_1 = \mathbf{U}\mathbf{s} + \mathbf{e}_1 + p^{k-1}\cdot\mathbf{y} \in \mathbb{Z}_q^{2\ell}.$$
Then, return the ciphertext $C = (\mathbf{c}_0, \mathbf{c}_1)$.
- $\mathcal{FE}.\mathsf{Dec}(\mathsf{sk}_\mathsf{id}, C)$. Given $C = (\mathbf{c}_0, \mathbf{c}_1)$ and a secret key $(\bar{\mathbf{x}}, \mathbf{z}_\mathbf{x})$ for $\mathbf{x} \in \mathbb{Z}_p^{2\ell}$, compute $\mu' = \langle \bar{\mathbf{x}}, \mathbf{c}_1\rangle - \langle \mathbf{z}_\mathbf{x}, \mathbf{c}_0\rangle \bmod q$ and output the value $\mu \in \mathbb{Z}_p$ that minimizes $|p^{k-1}\cdot\mu - \mu'|$.

Setting the Parameters. We have to set the parameters so that the correctness requirement is satisfied and the security reduction from $\mathrm{LWE}_{n,m,q,\alpha'}$ works, for some non-trivial error rate α'. We require that

- $p^{k-1}/4 > \sigma_0(\sigma_1+\sigma_2)\alpha q\sqrt{\ell m}\cdot\omega(\log^{3/2}\lambda)$, to ensure that the error term in decryption has magnitude less than $p^{k-1}/4$ with probability $1 - \lambda^{-\omega(1)}$,
- $\sigma_1, \sigma_2 \geq p\cdot\Omega(\sqrt{\lambda})$, to be able to apply Lemma 4 in the security proof,
- $\alpha/\alpha' \geq \Omega(\sigma_0\sqrt{m})$ and $\sigma_0, \alpha' q \geq \Omega(\sqrt{\lambda})$, to be able to apply Lemma 5 in the security proof,
- $\kappa \geq \Omega(\lambda + L\log\lambda)$, to ensure the (overwhelmingly likely) existence of a $\mathbf{U}$ as in Lemma 9 in the security proof,
- $\sigma_1 \geq \Omega(\sqrt{\ell\kappa\log\ell})$, $\ell \geq \Omega(\kappa\log(\sigma_1\kappa))$, and $\sigma_2 \geq \Omega(\kappa^{5/2}\sqrt{\ell}\sigma_1^2\log^{3/2}(\ell\sigma_1))$, to be able to apply Lemma 9 in the security proof with $\kappa \geq \Omega(\lambda + L\log\lambda)$ as above,
- $\sigma_0 \geq \Omega(p\kappa\ell\sigma_2)$ and $q^{n+1}/p^m \leq 2^{-\Omega(\kappa)}$, to be able to apply Lemma 19 in the security proof.

To satisfy the above requirements and rely on LWE parameters for which all known attacks cost $2^{o(\lambda)}$, we may set the parameters as follows. We choose $\kappa = \Theta(\lambda + L\log\lambda)$,

$p = \lambda^{\omega(1)}$, and:[7]

$$\begin{aligned} \ell &= \widetilde{\Theta}((\lambda + L)\log p) & \sigma_0 &= \widetilde{\Theta}((\lambda+L)^5 (p\log p)^3 \lambda) \\ \sigma_1 &= \Theta(p\sqrt{\lambda}) & \sigma_2 &= \widetilde{\Theta}((\lambda+L)^3 (p\log p)^2 \lambda) \\ 1/\alpha &= \widetilde{\Theta}((\lambda+L)^9 (p\log p)^6 \lambda^2) & 1/\alpha' &= \widetilde{\Theta}((\lambda+L)^{14.5} (p\log p)^9 \lambda^3) \\ m &= \widetilde{\Theta}(\lambda + L) & q &= \widetilde{\Theta}((\lambda+L)^{15} (p\log p)^9 \lambda^3) \\ k &= \Theta(1) & n &= \widetilde{\Theta}(\lambda) \end{aligned}$$

Decryption Correctness. To show the correctness of the scheme, we first observe that, modulo q:

$$\mu' = \langle \bar{\mathbf{x}}, \mathbf{c}_1 \rangle - \langle \mathbf{z}_{\mathbf{x}}, \mathbf{c}_0 \rangle = p^{k-1} \cdot \langle \mathbf{x}, \mathbf{y} \rangle + \langle \bar{\mathbf{x}}, \mathbf{e}_1 \rangle - \langle \mathbf{z}_{\mathbf{x}}, \mathbf{e}_0 \rangle.$$

Below, we show that the magnitude of the term $\langle \bar{\mathbf{x}}, \mathbf{e}_1 \rangle - \langle \mathbf{z}_{\mathbf{x}}, \mathbf{e}_0 \rangle$ is $\leq \sigma_0(\sigma_1 + \sigma_2)\alpha q\sqrt{\ell m} \cdot \omega(\log^{3/2}\lambda)$ with probability $1 - \lambda^{-\omega(1)}$. Thanks to the parameter choices, the latter upper bound is smaller than $p^{k-1}/4$, which suffices to guarantee decryption correctness.

Note that $\bar{x}_i \in \mathbb{Z}^{2\ell}$ is chosen from $D_{p\mathbb{Z}+x_i,\sigma_1}$ if $i \in [\ell]$ and $D_{p\mathbb{Z}+x_i,\sigma_2}$ otherwise. We thus have $\|\bar{\mathbf{x}}\| \leq (\sigma_1 + \sigma_2)\sqrt{\ell} \cdot \omega(\sqrt{\log \lambda})$ with probability $1 - \lambda^{-\omega(1)}$. This, together with $\mathbf{e}_1 \sim D^{2\ell}_{\mathbb{Z},\alpha q}$, implies that $|\langle \bar{\mathbf{x}}, \mathbf{e}_1 \rangle| \leq \alpha q(\sigma_1 + \sigma_2)\sqrt{\ell} \cdot \omega(\log \lambda)$ with probability $1 - \lambda^{-\omega(1)}$. Furthermore, since each column of $\mathbf{Z}$ is chosen from $D^{2\ell}_{\mathbb{Z},\sigma_0}$, we have $\|\mathbf{z}_{\mathbf{x}}\| \leq \sigma_0(\sigma_1 + \sigma_2)\sqrt{\ell m} \cdot \omega(\log \lambda)$ with probability $1 - \lambda^{-\omega(1)}$. As a result, we have $|\langle \mathbf{z}_{\mathbf{x}}, \mathbf{e}_0 \rangle| \leq \sigma_0(\sigma_1 + \sigma_2)\alpha q\sqrt{\ell m} \cdot \omega(\log^{3/2}\lambda)$ with probability $1 - \lambda^{-\omega(1)}$.

Security. We now show that the scheme above is secure, for our relaxed notion of L-IND-CPA security.

Theorem 18. *If the parameters are set as above, the above scheme is L-IND-CPA secure under the $\mathrm{LWE}_{n,m,q,\alpha'}$ assumption.*

Proof. The proof proceeds with a sequence of games that starts with the real game and ends with a game in which the adversary's advantage is negligible. For each i, we call S_i the event that the adversary wins in Game i.

Game 0: This is the ordinary security game. Namely, at the outset of the game, the adversary $\mathcal{A}$ is given the master public key pk. Then, it sees Q random vectors $\{\mathbf{x}_i\}_{i\in[Q]}$, where $\mathbf{x}_i \hookleftarrow \mathbb{Z}_p^{2\ell}$ and Q is an arbitrary polynomial specified by $\mathcal{A}$. Then, it makes secret key queries for these vectors. The number of the key queries is bounded by L. Note that the adversary can only make key queries for random vectors chosen as $\mathbf{x} \hookleftarrow \mathbb{Z}_p^{2\ell}$. In the challenge phase, the adversary $\mathcal{A}$ comes up with two distinct vectors $\mathbf{y}_0$, $\mathbf{y}_1$ and receives an encryption C of $\mathbf{y}_\beta$ for $\beta \hookleftarrow \{0,1\}$ sampled by the challenger. The adversary is not allowed to make secret key queries after the challenge phase. When $\mathcal{A}$ halts, it outputs $\beta' \in \{0,1\}$ and S_0 is the event that $\beta' = \beta$. Note that for any vector $\mathbf{x}$ for which $\mathcal{A}$ makes a secret key query, we must have $\langle \mathbf{x}, \mathbf{y}_0 \rangle \equiv \langle \mathbf{x}, \mathbf{y}_1 \rangle \bmod p$ if $\mathcal{A}$ is a legitimate adversary.

Game 1: We modify the generation of $\mathbf{x}$ and $\bar{\mathbf{x}}$ for all secret key queries. Namely, instead of choosing $\mathbf{x} \hookleftarrow \mathbb{Z}_p^{2\ell}$ and then sampling $\bar{\mathbf{x}}$, the challenger first chooses $\bar{\mathbf{x}} =$

[7] We note that it is possible to choose parameters that allow to take p as low as $p = 2$, but in our trace-and-revoke application we use $p = \lambda^{\omega(1)}$ to guarantee correctness.

$(\bar{x}_1, \ldots, \bar{x}_{2\ell})^t$ as $\bar{x}_i \hookleftarrow D_{\mathbb{Z},\sigma_1}$ for $i \in [\ell]$ and $\bar{x}_i \hookleftarrow D_{\mathbb{Z},\sigma_2}$ for $i \in [\ell+1, 2\ell]$ and then sets $\mathbf{x} := \bar{\mathbf{x}} \bmod p$. We claim that this changes the joint distribution of $(\mathbf{x}, \bar{\mathbf{x}})$ only negligibly. To see this, we observe that the distribution of $\bar{x}_i$ conditioned on $\bar{x}_i \equiv x_i \bmod p$ is $D_{p\mathbb{Z}+x_i,\sigma_1}$ for $i \in [\ell]$ and $D_{p\mathbb{Z}+x_i,\sigma_2}$ for $i \in [\ell+1, 2\ell]$. Therefore, it suffices to show that $\bar{x}_i \bmod p$ is statistically close to the uniform distribution over $\mathbb{Z}_p$ when $\bar{x}_i$ is chosen from $D_{\mathbb{Z},\sigma_1}$ or $D_{\mathbb{Z},\sigma_2}$. This follows from $\sigma_1, \sigma_2 \geq p \cdot \Omega(\sqrt{\lambda})$ and Lemma 4. Therefore, we have that $|\Pr[S_1] - \Pr[S_0]| \leq 2^{-\Omega(\lambda)}$.

Game 2: We modify the generation of $C = (\mathbf{c}_0, \mathbf{c}_1)$ in the challenge phase. Namely at the outset of the game, the challenger picks $\mathbf{s} \hookleftarrow \mathbb{Z}_q^n$, $\mathbf{e} \hookleftarrow D^m_{\mathbb{Z},\alpha' q}$ (which may be chosen ahead of time) as well as $\mathbf{Z} \hookleftarrow D^{2\ell \times m}_{\mathbb{Z},\sigma_0}$. Let $\mathbf{V} \in \mathbb{Z}^{(m+2\ell)\times m}$ be the matrix that is obtained by putting $\mathbf{I}_m$ on top of $\mathbf{Z}$, where $\mathbf{I}_m$ is the unit matrix of size m. We then set the ciphertext $C = (\mathbf{c}_0, \mathbf{c}_1) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{2\ell}$ as

$$\begin{aligned} \mathbf{b} &= \mathbf{A}\mathbf{s} + \mathbf{e} \\ (\mathbf{c}_0 \| \mathbf{c}_1) &= \mathsf{ReRand}(\mathbf{V}, \mathbf{b}, \alpha' q, \alpha/(2\alpha')) + p^{k-1} \cdot \mathbf{y}_\beta \end{aligned} \tag{4}$$

where ReRand is from Lemma 5. We claim that this change alters the view of the adversary only negligibly. To show this, we first observe that $s_1(\mathbf{V}) \leq \sqrt{1 + s_1(\mathbf{Z})^2} \leq O(\sigma_0 \sqrt{m})$ holds with all but $2^{-\Omega(m)} \leq 2^{-\Omega(\lambda)}$ probability by Lemma 7. By Lemma 5 and our parameter choices, we have

$$\begin{aligned} \mathbf{c}_0 &= \mathbf{I}_m \cdot \mathbf{A}\mathbf{s} + \mathbf{e}_0 = \mathbf{A}\mathbf{s} + \mathbf{e}_0, \\ \mathbf{c}_1 &= \mathbf{Z} \cdot \mathbf{A}\mathbf{s} + \mathbf{e}_1 + p^{k-1} \cdot \mathbf{y}_\beta = \mathbf{U}\mathbf{s} + \mathbf{e}_1 + p^{k-1} \cdot \mathbf{y}_\beta, \end{aligned}$$

where $\mathbf{e}_0$ and $\mathbf{e}_1$ are within statistical distance $2^{-\Omega(\lambda)}$ from $D^m_{\mathbb{Z},\alpha q}$. Therefore,we have that $|\Pr[S_2] - \Pr[S_1]| \leq 2^{-\Omega(\lambda)}$.

Game 3: We further modify the generation of $C = (\mathbf{c}_0, \mathbf{c}_1)$ in the challenge phase. Instead of setting $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$, we choose $\mathbf{b} = \mathbf{u} + \mathbf{e}$, where $\mathbf{u} \hookleftarrow \mathbb{Z}_q^m$. Then, the ciphertext is set as in Equation (4). Under the LWE assumption, we have that $|\Pr[S_3] - \Pr[S_2]|$ is negligible.

Game 4: We modify the generation of $C = (\mathbf{c}_0, \mathbf{c}_1)$ once more. Namely, the ciphertext is now set as

$$\begin{aligned} \mathbf{c}_0 &= \mathbf{u} + \mathbf{e}_0, \\ \mathbf{c}_1 &= \mathbf{Z} \cdot \mathbf{u} + \mathbf{e}_1 + p^{k-1} \cdot \mathbf{y}_\beta, \end{aligned}$$

where $\mathbf{u} \hookleftarrow \mathbb{Z}_q^m$ and $\mathbf{e}_0, \mathbf{e}_1 \hookleftarrow D^m_{\mathbb{Z},\alpha q}$. Similarly to Game 2, this change does not alter the view of the adversary much. By Lemma 5 and our parameter choices, we have that $|\Pr[S_4] - \Pr[S_3]| \leq 2^{-\Omega(\lambda)}$. Below, we prove that $\Pr[S_4]$ is exponentially close to 1/2, which will complete the proof.

Define $\mathbf{y} = \mathbf{y}_1 - \mathbf{y}_0 \in \mathbb{Z}_p^{2\ell}$. Let $\{\mathbf{x}_{i_j} \in \mathbb{Z}_p^{2\ell}\}_{j \in [L']}$ be the vectors corresponding to the secret key queries made by $\mathcal{A}$, where $L' \leq L$. As $\mathcal{A}$ is a legitimate adversary, we have $\langle \mathbf{x}_{i_j}, \mathbf{y} \rangle = 0 \bmod p$ for each secret key query $\mathbf{x}_{i_j}$. The view of the adversary contains L' tuples $\{\mathbf{x}_{i_j}, \bar{\mathbf{x}}_{i_j}, \mathbf{z}_{\mathbf{x}_{i_j}}\}_{j \in [L']}$, where the vectors $\{\mathbf{x}_{i_j}\}_{j \in [L']}$ form a $\mathbb{Z}_p$-basis of a subspace of the $(2\ell-1)$-dimensional vector space $\mathbf{y}^\perp := \{\mathbf{x} \in \mathbb{Z}_p^{2\ell} : \langle \mathbf{x}, \mathbf{y} \rangle = 0 \bmod p\}$. We define $\mathbf{X}_{top} \in \mathbb{Z}^{L' \times 2\ell}$ as the matrix whose j-th row is $\bar{\mathbf{x}}^t_{i_j}$ for $j \in [L']$.

We say that $\mathbf{X}_{top} \in \mathbb{Z}^{L' \times 2\ell}$ is *good* when we can choose $\mathbf{U} \in \mathbb{Z}^{2\ell \times 2\ell}$ such that $|\det \mathbf{U}| = 1$, $\mathbf{X}_{top} \cdot \mathbf{U} = (\mathbf{I}_{L'} | \mathbf{0})$, and every row of $\mathbf{U}$ has norm $\leq O(\sqrt{\kappa \ell} \sigma_2)$ (see Lemma

9). For a good $\mathbf{X}_{top}$, we can define $\mathbf{X} \in \mathbb{Z}^{2\ell \times 2\ell}$ as $\mathbf{X} := \mathbf{U}^{-1}$. It can be seen that the upper L' rows of $\mathbf{X}$ corresponds to $\mathbf{X}_{top}$. We denote the lower $2\ell - L'$ rows of the matrix as $\mathbf{X}_{bot}$. We note that since $\mathbf{X}$ is invertible over $\mathbb{Z}$, so is it modulo q. Without loss of generality, we assume that $\mathbf{U}$ and $\mathbf{X}_{bot}$ are deterministically determined from $\mathbf{X}_{top}$. (If there are more than one matrix satisfying the required properties, we sort them in the lexicographical order and pick the first one.) Note that it might be infeasible to efficiently compute $\mathbf{U}$ and $\mathbf{X}_{bot}$ from $\mathbf{X}_{top}$. This does not cause any problem in our proof because all the following arguments are information theoretic.

We state the following lemmas:

Lemma 19. *Assume that $\sigma_0 \geq \Omega(p\kappa\ell\sigma_2)$ and $q^{n+1}/p^m \leq 2^{-\Omega(\kappa)}$. Then the following distributions are within $2^{-\Omega(\kappa)}$ statistical distance:*

$$(\mathbf{A}, \mathbf{u}, \mathbf{ZA}, \mathbf{X}_{top}, \mathbf{X}_{top}\mathbf{Z}, \mathbf{X}_{bot}\mathbf{Zu}) \approx (\mathbf{A}, \mathbf{u}, \mathbf{ZA}, \mathbf{X}_{top}, \mathbf{X}_{top}\mathbf{Z}, \mathbf{v})$$

where $\mathbf{A} \hookleftarrow \mathbb{Z}_q^{m \times n}$, $\mathbf{u} \hookleftarrow \mathbb{Z}_q^m$, $\mathbf{Z} \hookleftarrow D_{\mathbb{Z},\sigma_0}^{2\ell \times m}$, each row of $\mathbf{X}_{top} \in \mathbb{Z}_q^{L' \times 2\ell}$ is chosen from $D_{\mathbb{Z},\sigma_1}^{\ell} \times D_{\mathbb{Z},\sigma_2}^{\ell}$, and $\mathbf{v} \hookleftarrow \mathbb{Z}_q^{2\ell - L'}$. Note that if $\mathbf{X}_{bot}$ is not good, then $\mathbf{X}_{bot}$ is undefined. In such a case, the term $\mathbf{X}_{bot}\mathbf{Zu}$ is replaced with $\perp$.

Lemma 20. *If there exists an adversary $\mathcal{A}$ whose advantage in **Game 4** is ϵ, then there exists another (unbounded) adversary $\mathcal{B}$ whose distinguishing advantage between the two distributions in Lemma 19 is $\varepsilon/Q^{L'}$.*

Given these two lemmas, we can conclude the proof of Theorem 18 since these imply $\varepsilon/Q^{L'} < 2^{-\Omega(\kappa)}$ and thus $\varepsilon < Q^L \cdot 2^{-\Omega(\kappa)} = 2^{O(L \log \lambda) - \Omega(\kappa)} \leq 2^{-\Omega(\lambda)}$. □

It remains to prove Lemmas 19 and 20.

Proof (of Lemma 19). By Lemma 9, matrix $\mathbf{X}_{top}$ is good with all but $2^{-\Omega(\kappa)}$ probability. In the following, let us fix good $\mathbf{X}_{top}$ and prove that the above two distributions are $2^{-\Omega(\kappa)}$-close. We first consider the distribution $\mathbf{X}_{bot}\mathbf{Z}$ conditioned on $\mathbf{X}_{top}\mathbf{Z}$. Note that in $\mathbf{X}_{top}\mathbf{Z}$ and $\mathbf{X}_{bot}\mathbf{Z}$, matrices $\mathbf{X}_{top}$ and $\mathbf{X}_{bot}$ act in parallel on the columns of $\mathbf{Z}$. We can hence restrict ourselves to the distribution of $\mathbf{X}_{bot}\mathbf{z}_i$ conditioned on $\mathbf{X}_{top}\mathbf{z}_i$, with $\mathbf{z}_i$ sampled from $D_{\mathbb{Z}^{2\ell},\sigma_0}$. Let $\mathbf{b}_i = \mathbf{X}_{top}\mathbf{z}_i \in \mathbb{Z}^{L'}$ and fix $\mathbf{z}_i^\star \in \mathbb{Z}^{2\ell}$ arbitrary such that $\mathbf{b}_i = \mathbf{X}_{top}\mathbf{z}_i^\star$. The distribution of $\mathbf{z}_i$ given $(\mathbf{X}_{top}, \mathbf{X}_{top}\mathbf{z}_i)$ is $D_{\Lambda + \mathbf{z}_i^\star, \sigma_0}$, with $\Lambda = \{\mathbf{x} \in \mathbb{Z}^{2\ell} : \mathbf{X}_{top}\mathbf{x} = \mathbf{0}\}$. Therefore, we have that given $\mathbf{X}_{top}\mathbf{z}_i$, the vector $\mathbf{X}_{bot}\mathbf{z}_i$ is distributed as $\mathbf{X}_{bot} \cdot D_{\Lambda + \mathbf{z}_i^\star, \sigma_0}$.

Let $\mathbf{U}_{lef}$ (resp. $\mathbf{U}_{rig}$) be the left L' (resp. right $2\ell - L'$) columns of $\mathbf{U}$. We show that the distribution $\mathbf{X}_{bot} \cdot D_{\Lambda + \mathbf{z}_i^\star, \sigma_0}$ actually corresponds to $D_{\mathbb{Z}^{2\ell - L'}, \sigma_0 \sqrt{\mathbf{\Sigma}}^{-1}, \mathbf{w}}$, where $\mathbf{\Sigma} = \mathbf{U}_{rig}^t \mathbf{U}_{rig}$ and $\mathbf{w} = -\sqrt{\mathbf{\Sigma}}^{-t} \mathbf{U}_{rig}^t \mathbf{U}_{lef} \mathbf{b}_i$. To see this, we first show that the supports of both distributions are the same. More specifically, we prove $\mathbf{X}_{bot} \cdot \Lambda = \mathbb{Z}^{2\ell - L'}$. To do so, it suffices to show that for any $\mathbf{a} \in \mathbb{Z}^{2\ell - L'}$, we have $\mathbf{a} \in \mathbf{X}_{bot} \cdot \Lambda$. By the construction of $\mathbf{U}$, we have $\mathbf{X}_{top}\mathbf{U}_{rig} = \mathbf{0}$ and $\mathbf{X}_{bot}\mathbf{U}_{rig} = \mathbf{I}_{2\ell - L'}$. Now, $\mathbf{a} \in \mathbf{X}_{bot} \cdot \Lambda$ follows because we have $\mathbf{a} = \mathbf{X}_{bot} \cdot (\mathbf{U}_{rig}\mathbf{a})$ and $\mathbf{X}_{top} \cdot \mathbf{U}_{rig}\mathbf{a} = \mathbf{0}$. We next evaluate the probability of $\mathbf{a} \in \mathbb{Z}^{2\ell - L'}$ being output by $\mathbf{X}_{bot} \cdot D_{\Lambda + \mathbf{z}_i^\star, \sigma_0}$. This probability equals to the probability of $\mathbf{a}' \in \mathbb{Z}^{2\ell}$ being output by $D_{\Lambda + \mathbf{z}_i^\star, \sigma_0}$ for $\mathbf{a}'$ that is the unique vector in $\Lambda + \mathbf{z}_i^\star$ satisfying $\mathbf{a} = \mathbf{X}_{bot} \cdot \mathbf{a}'$. Since $\mathbf{a}' \in \Lambda + \mathbf{z}_i^\star$, we have $\mathbf{X}_{top} \cdot \mathbf{a}' = \mathbf{X}_{top}(\mathbf{a}' - \mathbf{z}_i^\star) + \mathbf{X}_{top}\mathbf{z}_i^\star = \mathbf{0} + \mathbf{b}_i = \mathbf{b}_i$. Therefore, the vector $\mathbf{a}'$ can be written as $\mathbf{a}' = \mathbf{X}^{-1}(\mathbf{b}_i \| \mathbf{a}) = \mathbf{U}(\mathbf{b}_i \| \mathbf{a}) = \mathbf{U}_{lef}\mathbf{b}_i +$

$\mathbf{U}_{rig}\mathbf{a}$. The probability we consider is proportional to

$$\begin{aligned}\exp\left(-\pi\|\mathbf{a}'\|^2/\sigma_0^2\right) &= \exp\left(-\pi\|\mathbf{U}_{lef}\mathbf{b}_i + \mathbf{U}_{rig}\mathbf{a}\|^2/\sigma_0^2\right)\\ &= \exp\left(-\pi\|\sqrt{\mathbf{\Sigma}}\mathbf{a} + \sqrt{\mathbf{\Sigma}}^{-t}\mathbf{U}_{rig}^t\mathbf{U}_{lef}\mathbf{b}_i\|^2/\sigma_0^2\right)\\ &\quad\cdot\underbrace{\exp\left(-\pi(\|\mathbf{U}_{lef}\mathbf{b}_i\|^2 - \|\sqrt{\mathbf{\Sigma}}^{-t}\mathbf{U}_{rig}^t\mathbf{U}_{lef}\mathbf{b}_i\|^2)/\sigma_0^2\right)}_{\text{does not depend on }\mathbf{a}}\\ &\propto \exp\left(-\pi\|\sqrt{\mathbf{\Sigma}}(\mathbf{a}-\mathbf{w})\|^2/\sigma_0^2\right).\end{aligned}$$

This implies this equals to the probability of $\mathbf{a}$ being output by $D_{\mathbb{Z}^{2\ell-L'},\sigma_0\sqrt{\mathbf{\Sigma}}^{-1},\mathbf{w}}$. To sum up, conditioned on $\mathbf{X}_{top}\mathbf{Z}$, the matrix $\mathbf{X}_{bot}\mathbf{Z}$ is distributed as $(D_{\mathbb{Z}^{2\ell-L'},\sigma_0\sqrt{\mathbf{\Sigma}}^{-1},\mathbf{w}})^m$.

We then consider the joint distribution of $(\mathbf{A}, \mathbf{u}, \mathbf{ZA}, \mathbf{X}_{bot}\mathbf{Zu})$ (conditioned on $(\mathbf{X}_{top}, \mathbf{X}_{top}\mathbf{Z})$). In the following, let us consider the distribution of $\mathbf{XZA}$ instead of $\mathbf{ZA}$. We do not loose any information by doing this since $\mathbf{X}$ is invertible modulo q and the latter distribution can be recovered from the former by just multiplying $\mathbf{X}^{-1}$ from the left. Furthermore, we observe that $\mathbf{XZA}$ is the vertical concatenation of $\mathbf{X}_{top}\mathbf{ZA}$ and $\mathbf{X}_{bot}\mathbf{ZA}$. Since the former can be recovered from $\mathbf{X}_{top}\mathbf{Z}$ and $\mathbf{A}$, which are already included in the tuples, we ignore the former.

Let us denote $\mathbf{Y} := \mathbf{X}_{bot}\mathbf{Z} \sim (D_{\mathbb{Z}^{2\ell-L'},\sigma_0\sqrt{\mathbf{\Sigma}}^{-1},\mathbf{w}})^m$. To complete the proof, we will show that the following distributions are statistically close:

$$(\mathbf{A}, \mathbf{u}, \mathbf{YA}, \mathbf{Yu}) \approx (\mathbf{A}, \mathbf{u}, \mathbf{B}, \mathbf{v})$$

where $\mathbf{B} \hookleftarrow \mathbb{Z}_q^{(2\ell-L')\times n}$, and $\mathbf{v} \hookleftarrow \mathbb{Z}_q^{2\ell-L'}$. We first show that $(\mathbf{Y} \bmod p)$ is within $2^{\Omega(-\lambda)}$ statistical distance from the uniform distribution over $\mathbb{Z}_p^{(2\ell-L')\times m}$. This follows by setting $\Lambda = \mathbb{Z}^{2\ell-L'}$ and $\Lambda' = p\cdot\mathbb{Z}^{2\ell-L'}$ and applying Lemma 4 to $\mathbf{Y}$ in a column-wise manner. We check that the parameters satisfy the required condition of Lemma 4. We have

$$s_{2\ell-L'}(\sqrt{\mathbf{\Sigma}}^{-1}) = s_1(\mathbf{\Sigma})^{-1/2} \geq ((2\ell-L')^2\cdot\|\mathbf{\Sigma}\|_\infty)^{-1/2} \geq \Omega((\kappa^{1/2}\ell\sigma_2)^{-1}),$$

where the last inequality follows from the upper bound on the norms of the rows of $\mathbf{U}$. We therefore have $\sigma_0\cdot s_{2\ell-L'}(\sqrt{\mathbf{\Sigma}}^{-1}) \geq p\cdot\Omega(\sqrt{\kappa})$ by our choice of σ_0. We then finally apply Lemma 6 in a row-wise manner to obtain that $\mathbf{Y}(\mathbf{A}|\mathbf{u})$ is almost uniformly random. We note that the lemma can be applicable because $q^{n+1}/p^m \leq 2^{-\Omega(\kappa)}$. This completes the proof of Lemma 19. □

Proof (of Lemma 20). The reduction works as follows. Given $(\mathbf{A}, \mathbf{u}, \mathbf{ZA}, \mathbf{X}_{top}, \mathbf{X}_{top}\mathbf{Z}, \mathbf{v})$, algorithm $\mathcal{B}$ randomly guesses indices $\{i_j\}_{j\in L'} \in [Q]^{L'}$ for which the adversary makes key queries. The public key $\mathsf{pk} = (\mathbf{A}, \mathbf{U} = \mathbf{Z}\cdot\mathbf{A})$ and the master key $\mathsf{msk} = \mathbf{Z}$ are determined by the given problem instance. (Note that $\mathbf{Z}$ is not given to $\mathcal{B}$, so it is implicitly chosen.) Then $\mathcal{B}$ chooses $\{\bar{\mathbf{x}}_i\}_{i\in[Q]}$ as follows. When $i \in \{i_j\}_{j\in[L']}$, there exists j such that $i = i_j$. Then algorithm $\mathcal{B}$ sets $\bar{\mathbf{x}}_i^t$ to be the j-th row of the given matrix $\mathbf{X}_{top}$. Otherwise, it chooses $\bar{\mathbf{x}}_i$ as in **Game 4**. The key queries are handled as follows. Whenever $\mathcal{A}$ queries key for $\mathbf{x}_i$ such that $i \notin \{i_j\}_{j\in[L']}$, algorithm $\mathcal{B}$ aborts and outputs a random bit. Other queries can be handled using $\mathbf{X}_{top}\mathbf{Z}$ in the problem instance. To create the challenge ciphertext $\mathcal{B}$ sets

$$\mathbf{c}_0 = \mathbf{u}, \qquad \mathbf{c}_1 = \mathbf{X}^{-1}\cdot(\mathbf{X}_{top}\mathbf{Zu}\|\mathbf{v}) + \mathbf{e}_1 + \mathbf{y}_\beta.$$

We can observe that when $\mathbf{v} = \mathbf{X}_{bot}\mathbf{Zu}$, we have $\mathbf{c}_1 = \mathbf{Zu}+\mathbf{e}_1+\mathbf{y}_\beta$ and the distribution of the challenge ciphertext corresponds to that of **Game 4**.

We then consider the case of $\mathbf{v}$ is random. We will show that the distribution of $\mathbf{X} \cdot \boldsymbol{c}_1 \bmod q$ is independent of β. As the matrix $\mathbf{X}$ is independent of $\beta \in \{0, 1\}$ and invertible over $\mathbb{Z}_q$, this implies that the distribution of $\mathbf{c}_1$ is independent of β as well (recall that $\mathbf{X}$ is information theoretically known to $\mathcal{A}$, which means that, if $\mathbf{c}_1$ carries any information on β, so does $\mathbf{X} \cdot \mathbf{c}_1 \bmod q$). The first L' entries of $\mathbf{X} \cdot \mathbf{c}_1$ (namely, $\mathbf{X}_{top} \cdot \mathbf{c}_1$) do not depend on β because we have the equality $p^{k-1} \cdot \mathbf{X}_{top} \cdot \mathbf{y}_0 = p^{k-1} \cdot \mathbf{X}_{top} \cdot \mathbf{y}_1 \bmod q$, by construction of $\mathbf{X}_{top}$. The last $2\ell - L'$ entries are uniformly random, since they are masked by the random vector $\mathbf{v}$.

At the end of the game, algorithm $\mathcal{B}$ outputs the same bit as $\mathcal{A}$.

It can be seen that $\mathcal{B}$ perfectly simulates **Game 4** when $\mathbf{v} = \mathbf{X}_{bot}\mathbf{Zu}$ and a game that is independent of β when $\mathbf{v}$ is random. Therefore, conditioned on $\mathcal{B}$ not aborting, the distinguishing advantage of $\mathcal{B}$ is the same as $\mathcal{A}$. Since $\mathcal{B}$ aborts and outputs a random bit with probability $1/Q^{L'}$, the advantage of $\mathcal{B}$ is $\epsilon/Q^{L'}$. This completes the proof of Lemma 20. □

5 Trace and Revoke from DDH and Paillier

In this section, we describe two (near) instantiations of the generic construction presented in the last section. We are not aware of existing IPFE schemes that allow black-box instantiations, but some existing ones can be made to fit the framework.

5.1 Trace and Revoke from DDH

Following the work of Abdalla *et al.* [1], two DDH-based adaptively secure IPFEs modulo the group size q have been proposed [4, 8]. However, these schemes enjoy limited correctness: as decryption involves the computation of a discrete logarithm, one restricts the set of exponents to be small. For instance, one may design the schemes so that inner products that are small compared to q can be decrypted. This restriction seems incompatible with the requirements of our trace-and-revoke scheme, as the inner product $m \cdot \langle \mathbf{x}_{\mathsf{id}}, \mathbf{v}_{\mathcal{R}} \rangle$ occurring in the decryption algorithm has no reason to be small compared to p, even if the plaintext m is small. In the DDH-based trace-and-revoke scheme below, we circumvent the issue for the scheme from [4] by removing the $\langle \mathbf{x}_{\mathsf{id}}, \mathbf{v}_{\mathcal{R}} \rangle$ component before taking the discrete logarithm.

- $\mathsf{Setup}(1^\lambda, 1^t, 1^r, L)$. Choose a cyclic group $\mathbb{G}$ of prime order q along with two generators $g, h \hookleftarrow \mathbb{G}$. DDH in $\mathbb{G}$ should be 2^λ-hard, but taking base-g logarithms of elements g^x with $x \in \{1, \ldots, L\}$ should be tractable. Set $\ell = t + r + 1$. For each $i \leq \ell$, sample $s_i, t_i \hookleftarrow \mathbb{Z}_q$ and compute $h_i = g^{s_i} \cdot h^{t_i}$. Define
$$\mathsf{msk} := (\mathbf{s}, \mathbf{t}) \ \text{ and } \ \mathsf{pk} := \big(\mathbb{G}, g, h, \{h_i\}_{i \in [\ell]}\big).$$
- $\mathsf{KeyGen}(\mathsf{msk}, \mathsf{id})$. Sample $\mathbf{x}_{\mathsf{id}} \hookleftarrow \mathbb{Z}_q^\ell$. Set $\mathsf{sk}_{id} = (\langle \mathbf{x}_{\mathsf{id}}, \mathbf{s} \rangle, \langle \mathbf{y}_{\mathsf{id}}, \mathbf{s} \rangle) \in \mathbb{Z}_q^2$ and $p_{\mathsf{id}} = \mathbf{x}_{\mathsf{id}}$.
- $\mathsf{Enc}(\mathsf{pk}, \mathcal{R}, m)$ proceeds as follows to encrypt $m \in \{1, \ldots, L\}$.
  1. Compute $\mathbf{v}_{\mathcal{R}} \in \mathbb{Z}_q^\ell \setminus \{\mathbf{0}\}$ such that $\langle \mathbf{x}_{\mathsf{id}}, \mathbf{v}_{\mathcal{R}} \rangle = 0$ for every $\mathsf{id} \in \mathcal{R}$.
  2. Set $\mathbf{y} = m \cdot \mathbf{v}_{\mathcal{R}}$ and sample $r \hookleftarrow \mathbb{Z}_q$.
  3. Compute $D_0 = g^r$, $D_1 = h^r$ and $E_i = g^{y_i} \cdot h_i^r$ for all $i \leq \ell$.

  The ciphertext C is $(D_0, D_1, E_1, \ldots, E_\ell, \mathcal{R})$.

- $\mathsf{Dec}(\mathsf{sk}_{\mathsf{id}}, C)$. Write $C = (D_0, D_1, E_1, \ldots, E_\ell, \mathcal{R})$ and let $\mathbf{x}_{\mathsf{id}}$ denote the vector corresponding to $\mathsf{sk}_{\mathsf{id}} = (s_x, t_x)$. Compute:

$$C_{\mathbf{x}_{\mathsf{id}}} = \Big(\prod_{i=1}^{\ell} E_i^{x_{\mathsf{id},i}}\Big)/(D_0^{s_x} \cdot D_1^{t_x}).$$

  Then, recover $\mathbf{v}_{\mathcal{R}}$ from $\mathcal{R}$ and output the base-g logarithm of $C_{\mathbf{x}_{\mathsf{id}}}^{1/\langle \mathbf{v}_{\mathcal{R}}, \mathbf{x}_{\mathsf{id}}\rangle}$.
- $\mathsf{Trace}(\mathsf{pd}, \mathcal{S}, \mathcal{R}, \mathcal{O}^{\mathcal{D}})$ proceeds as described in Section 3.

Correctness follows from elementary computations. The only difference with the direct instantiation of our trace-and-revoke construction is that the division by $\langle \mathbf{v}_{\mathcal{R}}, \mathbf{x}_{\mathsf{id}}\rangle$ occurs before the computation of the discrete logarithm, hence enabling efficient decryption.

Key and Ciphertext Sizes. Both the public key and master secret key consist of $O((t+r)\log q)$ bits. To every user id corresponds a secret key $\mathsf{sk}_{\mathsf{id}}$ of bit-length $O(\log q)$ and a vector p_{id} of bit-length $O((t+r)\log q)$. Algorithm Enc maps a plaintext in $\mathbb{Z}_q \setminus \{0\}$ to a ciphertext of bit-length $O((t+r)\log q)$. If we choose the DDH group as an elliptic curve group (without pairings), we may set $\log q = O(\lambda)$.

5.2 Trace and Revoke from Paillier

In [4], Agrawal *et al.* described two IPFEs relying on the algebraic framework of Paillier's encryption scheme [35]. One scheme handles inner products of short integers vectors, while the other handles inner products modulo a product $N = p \cdot q$ of two safe primes. Both are proved secure under the Decision Composite Residuosity (DCR) hardness assumption. We explain here how to instantiate our trace-and-revoke construction using this IPFE over $\mathbb{Z}_N^\ell$.

A first difficulty is the fact that the Key Generation algorithm is stateful. However, this issue can be handled by noticing that for random queries, the key generation algorithm can be made stateless (see [4] and Subsection 4.1 for more details). A further difficulty is the non-primality of N: our transformation requires the modulus to be prime. We may actually apply the transformation and "pretend" that N is prime, both in the scheme and in its security proof. The non-primality of N can be noticed only when finding vectors orthogonal modulo N to some specified vectors. When such a task is performed, either the linear algebra operations proceed and find such a vector, or they fail. In the latter case, a non-trivial factor of N has been found, which leads to an algorithm against DCR. Hence, under the DCR hardness assumption, such an event is unlikely. We now describe the resulting DCR-based trace-and-revoke scheme.

- $\mathsf{Setup}(1^\lambda, 1^t, 1^r)$. Choose safe prime numbers $p = 2p' + 1, q = 2q' + 1$ with sufficiently large primes $p', q' > 2^{\mathcal{P}oly(\lambda)}$, and compute $N = pq$. Then, sample $g' \hookleftarrow \mathbb{Z}_{N^2}^*$ and compute $g = g'^{2N} \bmod N^2$, which generates the subgroup of $(2N)$-th residues in $\mathbb{Z}_{N^2}^\star$ with overwhelming probability. Set $\ell = t + r + 1$ and sample $\mathbf{s}$ from the integer Gaussian distribution $D_{\mathbb{Z}^\ell,\sigma}$ with standard deviation parameter σ satisfying $\sigma \geq \sqrt{\ell} N \mathcal{P}oly(\lambda)$. Compute $h_i = g^{s_i} \bmod N^2$ for all $i \leq \ell$. Define

$$\mathsf{msk} := \mathbf{s} \ \text{ and } \ \mathsf{pk} := \big(N, g, \mathcal{G}, \{h_i\}_{i\in[\ell]}\big).$$

- $\mathsf{KeyGen}(\mathsf{msk}, \mathsf{id})$. Sample $\mathbf{x}_{\mathsf{id}} \in \mathbb{Z}^\ell$ with coefficients i.i.d. uniform in $\{0, \ldots, N-1\}$. Set $\mathsf{sk}_{id} = \langle \mathbf{x}_{\mathsf{id}}, \mathbf{s}\rangle \in \mathbb{Z}$ and $p_{\mathsf{id}} = \mathbf{x}_{\mathsf{id}}$.

- $\mathsf{Enc}(\mathsf{pk}, \mathcal{R}, m)$ proceeds as follows to encrypt $m \in \mathbb{Z}_N \setminus \{0\}$.
  1. Compute $\mathbf{v}_{\mathcal{R}} \in \mathbb{Z}_p^\ell \setminus \{\mathbf{0}\}$ such that $\langle \mathbf{x}_{\mathsf{id}}, \mathbf{v}_{\mathcal{R}} \rangle = 0$ for every $\mathsf{id} \in \mathcal{R}$.
  2. Set $\mathbf{y} = m \cdot \mathbf{v}_{\mathcal{R}}$ and sample $r \hookleftarrow \{0, \dots, \lfloor N/4 \rfloor\}$.
  3. Compute $C_0 = g^r \bmod N^2$ and $C_i = (1 + y_i N) \cdot h_i^r \bmod N^2$ for all $i \leq \ell$.

  The ciphertext C is $(C_0, C_1, \dots, C_\ell, \mathcal{R})$.
- $\mathsf{Dec}(\mathsf{sk}_{\mathsf{id}}, C)$. Write $C = (C_0, C_1, \dots, C_\ell, \mathcal{R})$ and let $\mathbf{x}_{\mathsf{id}}$ denote the vector corresponding to $\mathsf{sk}_{\mathsf{id}}$. Compute:

$$C_{\mathbf{x}_{\mathsf{id}}} = C_0^{-\mathsf{sk}_{\mathsf{id}}} \cdot \Big(\prod_{i=1}^{\ell} C_i^{x_{\mathsf{id},i}} \Big) \bmod N^2.$$

  Then, recover $\mathbf{v}_{\mathcal{R}}$ from $\mathcal{R}$ and output $(\frac{C_{\mathbf{x}_{\mathsf{id}}} - 1 \bmod N^2}{N}) / \langle \mathbf{v}_{\mathcal{R}}, \mathbf{x}_{\mathsf{id}} \rangle$.
- $\mathsf{Trace}(\mathsf{pd}, \mathcal{S}, \mathcal{R}, \mathcal{O}^{\mathcal{D}})$ proceeds as described in Section 3.

We note that by exploiting the fact that the attacker makes random queries, we may improve the parameter sizes provided by [4] exactly as in Subsection 4.2. In more detail, the proof of Theorem 5 (Appendix F) in [4] can be modified to show that the advantage of the adversary in Game 3 is negligible even when σ is chosen as above, exactly as described in Subsection 4.2.

Key and Ciphertext Sizes. The public key and master secret key respectively consist of $O((t+r)\log N)$ and $O((t+r)\log N)$ bits. Note that the master secret key bit-length can be shrunk to $O(\lambda)$ by only storing the seed of the pseudo-random generator used to create it. In that case, the master secret key may be recomputed every time the KeyGen algorithm is called. Further, to every user id corresponds a secret key $\mathsf{sk}_{\mathsf{id}}$ of bit-length $O((t+r)\log N)$ and a vector p_{id} of bit-length $O((t+r)\log N)$. Algorithm Enc maps a plaintext in $\mathbb{Z}_N \setminus \{0\}$ to a ciphertext of bit-length $O((t+r)\log N)$. To compensate for the number field sieve, we must choose $\log N = \widetilde{\Omega}(\lambda^3)$.

Acknowledgements. We thank Benoît Libert for helpful discussions. Sanjay Bhattacherjee was funded by the “Programme Avenir Lyon Saint-Etienne de l’Université de Lyon” in the framework of the programme “Investissements d’Avenir” (ANR-11-IDEX-0007). Damien Stehlé was supported by the ERC Starting Grant ERC-2013-StG-335086-LATTAC. Duong Hieu Phan was supported by the ANR ALAMBIC (ANR-16-CE39-0006). Shota Yamada was supported by JSPS KAKENHI Grant Number 16K16068 and JST CREST Grant Number JPMJCR1688, Japan.

References

1. M. Abdalla, F. Bourse, A. De Caro, and D. Pointcheval. Simple functional encryption schemes for inner products. In *PKC 2015*, LNCS, pages 733–751. Springer, 2015.
2. M. Abdalla, F. Bourse, A. De Caro, and D. Pointcheval. Better security for functional encryption for inner product evaluations. Cryptology ePrint Archive, Report 2016/011, 2016. `http://eprint.iacr.org/2016/011`.
3. S. Agrawal, C. Gentry, S. Halevi, and A. Sahai. Discrete Gaussian leftover hash lemma over infinite domains. In *Advances in Cryptology – ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 97–116. Springer, Dec. 2013.
4. S. Agrawal, B. Libert, and D. Stehlé. Fully secure functional encryption for inner products, from standard assumptions. In *Advances in Cryptology – CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 333–362. Springer, Aug. 2016.
5. S. Agrawal and A. Rosen. Functional encryption for bounded collusions, revisited. Eprint, 2016.
6. S. Bai, A. Langlois, T. Lepoint, D. Stehlé, and R. Steinfeld. Improved security proofs in lattice-based cryptography: Using the Rényi divergence rather than the statistical distance. In *Advances in Cryptology – ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 3–24. Springer, Dec. 2015.
7. B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. *J. ACM*, 59(2), May 2012.
8. F. Benhamouda, F. Bourse, and H. Lipmaa. CCA-secure inner-product functional encryption from projective hash functions. In *Proc. of PKC*, volume 10175 of *LNCS*, pages 36–66. Springer, 2017.
9. 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.
10. D. Boneh, A. Sahai, and B. Waters. Functional encryption: Definitions and challenges. In Y. Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 253–273. Springer, Mar. 2011.
11. D. Boneh and B. Waters. A fully collusion resistant broadcast, trace, and revoke system. In *Proc. of ACM CCS*, pages 211–220. ACM, 2006.
12. Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. Classical hardness of learning with errors. In *45th ACM STOC*, pages 575–584. ACM Press, 2013.
13. B. Chor, A. Fiat, and M. Naor. Tracing traitors. In *Proc. of CRYPTO*, volume 839 of *LNCS*, pages 257–270. Springer, 1994.
14. Y. Dodis and N. Fazio. Public key trace and revoke scheme secure against adaptive chosen ciphertext attack. In Y. Desmedt, editor, *PKC 2003*, volume 2567 of *LNCS*, pages 100–115. Springer, Jan. 2003.
15. A. Fiat and M. Naor. Broadcast encryption. In *Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '93, 1993.
16. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In R. E. Ladner and C. Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.
17. S. Goldwasser, Y. Tauman Kalai, R. Popa, V. Vaikuntanathan, and N. Zeldovich. Reusable garbled circuits and succinct functional encryption. In *Proc. of STOC*, pages 555–564. ACM Press, 2013.
18. S. Gorbunov, V. Vaikuntanathan, and H. Wee. Functional encryption with bounded collusions from multiparty computation. In *CRYPTO*, 2012.
19. S. Gorbunov, V. Vaikuntanathan, and H. Wee. Attribute-based encryption for circuits. In *45th ACM STOC*, pages 545–554. ACM Press, 2013.
20. D. Hofheinz and C. Striecks. A generic view on trace-and-revoke broadcast encryption schemes. In *CT-RSA 2014*, LNCS, pages 48–63. Springer, 2014.
21. S. Katsumata and S. Yamada. Partitioning via non-linear polynomial functions: More compact IBEs from ideal lattices and bilinear maps. In *Advances in Cryptology – ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 682–712. Springer, Dec. 2016.
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 L. R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 450–465. Springer, Apr. / May 2002.
25. C. H. Kim, Y. H. Hwang, and P. J. Lee. An efficient public key trace and revoke scheme secure against adaptive chosen ciphertext attack. In C.-S. Laih, editor, *ASIACRYPT 2003*, volume 2894 of *LNCS*, pages 359–373. Springer, Nov. / Dec. 2003.

26. K. Kurosawa and Y. Desmedt. Optimum traitor tracing and asymmetric schemes. In *Proc. of EUROCRYPT*, LNCS, pages 145–157. Springer, 1998.
27. K. Kurosawa and T. Yoshida. Linear code implies public-key traitor tracing. In D. Naccache and P. Paillier, editors, *PKC 2002*, volume 2274 of *LNCS*, pages 172–187. Springer, Feb. 2002.
28. A. Langlois, D. Stehlé, and R. Steinfeld. GGHLite: More efficient multilinear maps from ideal lattices. In *Proc. of EUROCRYPT*, LNCS, pages 239–256. Springer, 2014.
29. S. Ling, D. H. Phan, D. Stehlé, and R. Steinfeld. Hardness of k-LWE and applications in traitor tracing. In *Proc. of CRYPTO*, volume 8616 of *LNCS*, pages 315–334. Springer, 2014.
30. D. Micciancio and P. Mol. Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In P. Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 465–484. Springer, Aug. 2011.
31. D. Naor, M. Naor, and J. Lotspiech. Revocation and tracing schemes for stateless receivers. In J. Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 41–62. Springer, Aug. 2001.
32. M. Naor and B. Pinkas. Efficient trace and revoke schemes. In *Proc. of Financial Cryptography*, volume 1962 of *LNCS*, pages 1–20. Springer, 2000.
33. H. Q. Ngo, D. H. Phan, and D. Pointcheval. Black-box trace&revoke codes. *Algorithmica*, 67(3):418–448, 2013.
34. R. Nishimaki, D. Wichs, and M. Zhandry. Anonymous traitor tracing: how to embed arbitrary information in a key. In *Eurocrypt*, pages 388–419. Springer, 2016.
35. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In J. Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 223–238. Springer, May 1999.
36. C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In M. Mitzenmacher, editor, *41st ACM STOC*, pages 333–342. ACM Press, May / June 2009.
37. D. H. Phan and V. C. Trinh. Identity-based trace and revoke schemes. In X. Boyen and X. Chen, editors, *ProvSec 2011*, volume 6980 of *LNCS*, pages 204–221. Springer, Oct. 2011.
38. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In H. N. Gabow and R. Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.
39. A. Sahai and B. R. Waters. Fuzzy identity-based encryption. In R. Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, May 2005.
40. 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.
41. 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.

A Additional Relevant Work

There are multiple parameters in trace-and-revoke systems that one desires to optimize such as security definition, hardness assumption, public traceability, collusion size, efficiency. The most general adaptive security definition for trace and revoke was provided by Boneh and Waters [11]. Here, the adversary is permitted to adaptively make key requests, and must finally submit a pirate decoder. For the adversary to win the game, pirate decoder must be *useful*, i.e., the challenger must be allowed to test it with various "probe" ciphertexts and these must be decrypted with non-negligible probability, and the tracing algorithm must be able to output at least one user whose key was not requested by the adversary.

Strong Security for Trace and Revoke. The definition of usefulness of the pirate decoder involves a subtlety – in the strongest definition, the pirate decoder may be queried with ciphertexts that may encode a set of maliciously chosen revoked users [11]. Most constructions do not satisfy this strong notion of security, indeed some schemes are actually insecure in this strong game.

For instance in the schemes [32, 14], a probe ciphertext may be distinguished from a normal ciphertext using a revoked key. In the polynomial interpolation based method in [32], in order to run tracing on a suspect set, the authority chooses a probe polynomial which agrees with the original polynomial on all the points in the suspected set. Therefore, if the suspected set contains all the traitor keys, then the pirate cannot detect this change from the original polynomial to the probe polynomial and the tracing works well. However, if the pirate knows one key (an evaluation of the original polynomial) in the revoked set, then it can detect this change. This means that a revoked key is useless in decrypting ciphertexts but useful in detecting the presence of a tracing procedure. Therefore, the tracing algorithm from [32] does not allow the adversary to choose and corrupt keys of the revoked set in the tracing game.

Combinatorial Schemes We remark that another line of work constructs *combinatorial schemes* [13, 32, 40, 41, 31, 33], in contrast to the algebraic ones we have discussed so far; however these are usually less efficient than the algebraic candidates and the combination of trace and revoke is often studied in weaker security models.

Parameters Obtained with the NWZ Compiler. The NWZ compiler [34] may be instantiated with the bounded collusion functional encryption scheme from [18]. This results in a scheme that has a ciphertext size that depends polynomially on the size of the circuit used by NWZ, as well as quartically on the collusion bound $r + t$. Since the circuit used by NWZ has an input size of $O(r + t)$, the ciphertext size grows at least as $O((r + t)^5 \mathcal{P}oly(\lambda))$.

If the compiler is instantiated with the bounded collusion scheme of [17] (compiled with [18]), then the ciphertext size still grows as $O((r+t)^4 \mathcal{P}oly(\lambda))$, and moreover relies on the subexponential hardness of learning with errors in addition to heavy hammers such as fully homomorphic encryption and attribute based encryption. We note that the $\mathcal{P}oly(\lambda)$ factors above are unspecified, and possibly large: for instance, the circuit in [18] is represented using randomizing polynomials which adds a polynomial factor blow-up. Similarly, using the bounded collusion FE of [5] leads to better asymptotic bounds $O(r+t)^3 \mathcal{P}oly(\lambda))$ but also suffers from large polynomial factors, since again the circuit is represented using randomizing polynomials. Here, a quadratic factor $(r + t)^2$ is incurred by the query dependence of [5] and an additional factor $(r + t)$ is incurred due to circuit size dependence.

B Traitor Tracing for Multiple Messages

In this section, we provide a generic transformation from an inner product functional encryption scheme $\mathcal{FE}$ to a traitor tracing scheme $\mathcal{TT}$ that supports the encryption of multiple messages in one ciphertext. This allows us to obtain a relatively efficient publicly traceable traitor tracing scheme with security based on the DDH hardness assumption.

Let us first explain how the scheme works. Let k be the number of messages that each ciphertext encrypts and let t be the bound on the number of traitors. As was mentioned in Section 2.1, to get a traitor tracing scheme, revocation of users is no more considered in Definition 1 for $\mathcal{TR}$. As a result, the bound r is no more an input to the algorithms in $\mathcal{TR}$ and the set $\mathcal{R}$ is empty. In the AD-TT game between the challenger and the adversary $\mathcal{A}$, the adversary makes key queries only for the set $\mathcal{T}$ and not for the set $\mathcal{R}$.

Let $\mathcal{FE} = (\mathcal{FE}.\mathsf{Setup}, \mathcal{FE}.\mathsf{KeyGen}, \mathcal{FE}.\mathsf{Enc}, \mathcal{FE}.\mathsf{Dec})$ be an inner product functional encryption scheme over $\mathbb{Z}_p^\ell$ with $\ell = k + t$, as in Definition 2. The encryptor chooses a vector $\mathbf{m} = (m_1, \ldots, m_k)$ of messages and computes $C_\mathbf{m} \leftarrow \mathcal{FE}.\mathsf{Enc}(\mathsf{pk}, \mathbf{y})$ where $\mathbf{y} = (m_1, \ldots, m_k, 0, \ldots, 0) \in \mathbb{Z}_p^\ell$. As in the $\mathcal{TR}$ scheme, for key generation, a user with identity $\mathsf{id} \in \mathsf{ID}$ is first assigned a random t-dimensional vector $\mathbf{x}_{\mathsf{id}} \in \mathbb{Z}_p^t$.[8] The vector $\mathbf{x}_{\mathsf{id}}$ is used to construct k extended vectors $\hat{\mathbf{x}}_{\mathsf{id},1}, \dots, \hat{\mathbf{x}}_{\mathsf{id},k}$ (each of dimension ℓ) as:

$$\begin{aligned}
\hat{\mathbf{x}}_{\mathsf{id},1} &= (1, 0, \dots, 0, \mathbf{x}_{\mathsf{id}}) \\
\hat{\mathbf{x}}_{\mathsf{id},2} &= (0, 1, \dots, 0, \mathbf{x}_{\mathsf{id}}) \\
&\vdots \\
\hat{\mathbf{x}}_{\mathsf{id},k} &= (0, 0, \dots, 1, \mathbf{x}_{\mathsf{id}})
\end{aligned}$$

Every $\hat{\mathbf{x}}_{\mathsf{id},j}$ is used to compute the $\mathcal{FE}$ secret keys $\mathsf{sk}_{\mathsf{id}}^j$ for each $j \in [k]$. The user with identity id receives the secret key vector $\mathsf{sk}_{\mathsf{id}} = (\mathsf{sk}_{\mathsf{id}}^1, \dots, \mathsf{sk}_{\mathsf{id}}^k)$. We later verify (when proving the correctness of $\mathcal{TT}$) that $\mathcal{FE}$ decryption of $C_{\mathbf{m}}$ with key $\mathsf{sk}_{\mathsf{id}}^j$ yields the message component m_j.

In the AD-TT game, the decryption box $\mathcal{D}$ produced by the adversary $\mathcal{A}$ takes as input a ciphertext $C_{\mathbf{m}}$ and outputs the message vector $\tilde{\mathbf{m}} = (\tilde{m}_1, \dots, \tilde{m}_k)$. The oracle $\mathcal{O}^{\mathcal{D}}$ takes as input the message-ciphertext pair $(\mathbf{m}, C_{\mathbf{m}})$, and returns a k-tuple $\mathbf{b} = (b_1, \dots, b_k)$ of bits where

$$b_j = \begin{cases} 1 \text{ if } \tilde{m}_j = m_j, \\ 0 \text{ otherwise.} \end{cases}$$

We may consider the output bit b_j of $\mathbf{b} \leftarrow \mathcal{O}^{\mathcal{D}}(\mathbf{m}, C_{\mathbf{m}})$ as the output of an oracle ${\mathcal{O}^{\mathcal{D}}}_j(m_j, C_{\mathbf{m}})$ that indicates whether $\mathcal{D}$ decrypts the j^{th} component of $C_{\mathbf{m}}$ correctly. Given the oracle $\mathcal{O}^{\mathcal{D}}$, we may consider these component oracles ${\mathcal{O}^{\mathcal{D}}}_j$ for each $j \in [k]$ to be implicitly present.

A traitor with $\mathsf{id} \in \mathcal{T}$ may have leaked some or all of the k components of its secret key vector $\mathsf{sk}_{\mathsf{id}} = (\mathsf{sk}_{\mathsf{id}}^1, \dots, \mathsf{sk}_{\mathsf{id}}^k)$. The scheme is so designed that the leaked component $\mathsf{sk}_{\mathsf{id}}^j$ can only decrypt m_j correctly from $C_{\mathbf{m}}$ and not others. The decryption box $\mathcal{D}$ is considered to be useful for the component $j \in [k]$ if

$$\Pr_{\substack{\mathbf{m} \hookleftarrow U(\mathcal{M}^k) \\ C \hookleftarrow \mathsf{Enc}(\mathsf{pk}, \mathbf{m})}} \left[{\mathcal{O}^{\mathcal{D}}}_j(m_j, C) = 1 \right] \geq \frac{1}{|\mathcal{M}|} + \frac{1}{\lambda^{c_j}}, \tag{5}$$

for some constant $c_j > 0$. The decryption box $\mathcal{D}$ is considered to be useful if it is useful for at least one component $j \in [k]$.

The tracing algorithm starts with $j = 1$ and iterates for each component $j \leq k$ in an increasing order. The moment it finds a traitor for a component, it reports the corresponding user's id and aborts. For a component j under investigation, it starts by choosing two ℓ-dimensional vectors $\mathbf{y} = (m_1, \dots, m_j, \dots, m_k, 0, \dots, 0)$ and $\mathbf{y}' = (m_1, \dots, m_j', \dots, m_k, 0, \dots, 0)$, where $m_1, \dots, m_{j-1}, m_{j+1}, \dots, m_k \hookleftarrow \mathbb{Z}_p$ are chosen randomly and $m_j, m_j' \hookleftarrow \mathbb{Z}_p$ are chosen such that

$$\left| \Pr_{C \hookleftarrow \mathsf{Enc}(\mathsf{pk}, \mathbf{y})} \left[{\mathcal{O}^{\mathcal{D}}}_j(m_j, C) = 1 \right] - \Pr_{C' \hookleftarrow \mathsf{Enc}(\mathsf{pk}, \mathbf{y}')} \left[{\mathcal{O}^{\mathcal{D}}}_j(m_j, C') = 1 \right] \right| \geq \frac{1}{2\lambda^{c_j}}. \tag{6}$$

[8] As before, for full public traceability $p_{\mathsf{id}} = (\mathsf{id}, \mathbf{x}_{\mathsf{id}})$ is stored in pd while for almost public traceability only $p_{\mathsf{id}} = (\mathbf{x}_{\mathsf{id}})$ is stored in pd and $(\mathsf{id}, \mathbf{x}_{\mathsf{id}})$ is kept private by an authority. However, unlike the $\mathcal{TR}$ scheme, the encryption and decryption algorithms do not need access to the public directory pd. It is only required by the tracing algorithm Trace for computing the probe ciphertexts.

Given the set $\mathcal{S}$ of suspected traitors and some $\mathcal{S}_i \subseteq \mathcal{S}$, the algorithm chooses a vector $\mathbf{v}_{\mathcal{S}_i} \in \mathbb{Z}_p^\ell$ that satisfies for $j_1 \in [k]$:

$$\begin{aligned}
&\forall \mathsf{id} \in \mathcal{S}_i, \forall j_1 \in [k] : \langle \mathbf{v}_{\mathcal{S}}, \hat{\mathbf{x}}_{\mathsf{id},j_1} \rangle = 0 \bmod p,\\
&\qquad\qquad \text{and}\\
&\forall \mathsf{id} \in (\mathcal{S} \setminus \mathcal{S}_i) :\\
&\quad \text{if } j_1 \neq j : \qquad \langle \mathbf{v}_{\mathcal{S}}, \hat{\mathbf{x}}_{\mathsf{id},j_1} \rangle = 0 \bmod p,\\
&\quad \text{if } j_1 = j : \qquad \langle \mathbf{v}_{\mathcal{S}}, \hat{\mathbf{x}}_{\mathsf{id},j} \rangle = (m'_j - m_j) \bmod p.
\end{aligned}$$

A probe ciphertext C is constructed by computing an $\mathcal{FE}$ encryption of the vector $\mathbf{y} = \mathbf{v}_{\mathcal{S}_i} + (m_1, \ldots, m_j, \ldots, m_k, 0, \ldots, 0)$ and is submitted to the pirate decoder. The choice of m_j and m'_j ensures that the difference in probability of $\mathcal{O^D}_j(m_j, C) = 1$ for the probe ciphertexts C at the beginning and end of the jth iteration, is non-negligible. As in the $\mathcal{TR}$ scheme, this non-negligible difference in the probability helps in finding a traitor.

During the iteration for component j, if the decoder $\mathcal{D}$ is useful for a component $j_1 \neq j$, it decrypts that component j_1 correctly. Hence, usefulness of other components do not interrupt the investigation of component j.

B.1 The Scheme

In this section, we construct a traitor tracing scheme $\mathcal{TT}$ following Definition 1. Our scheme assumes the existence of a public directory pd which contains the identities of the users that have been assigned keys in the system. We assume that pd can only be modified by a central authority, i.e., the key generator.

1. $\mathsf{Setup}(1^\lambda, 1^k, 1^t)$. Upon input the security parameter λ, the size k of the message vector and the bound t on the number of traitors, proceed as follows:
   (a) Let $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathcal{FE}.\mathsf{Setup}(1^\lambda, 1^\ell)$, where $\ell = t + k$.
   (b) Output the public key pk and master secret key msk.
2. $\mathsf{Keygen}(\mathsf{msk}, \mathsf{id})$. Upon input the master secret key msk and a user identity $\mathsf{id} \in \mathsf{ID}$, proceed as follows:
   (a) Sample $\mathbf{x}_{\mathsf{id}} \hookleftarrow \mathbb{Z}_p^t$. The pair $p_{\mathsf{id}} = (\mathsf{id}, \mathbf{x}_{\mathsf{id}})$ is appended to the public directory pd.
   (b) Let $\mathbf{1}_j$ be a vector of length k with all its components set to 0 other than the jth component which is set to 1. For each $j \in [k]$, a vector $\hat{\mathbf{x}}_{\mathsf{id},j}$ of dimension $\ell = k + t$ is created from $\mathbf{x}_{\mathsf{id}}$ as
   $$\hat{\mathbf{x}}_{\mathsf{id},j} = (\mathbf{1}_j, \mathbf{x}_{\mathsf{id}}).$$
   (c) Let $\mathsf{sk}_{\mathsf{id}}^j \leftarrow \mathcal{FE}.\mathsf{KeyGen}(\mathsf{msk}, \hat{\mathbf{x}}_{\mathsf{id},j})$ for $j \in [k]$.
   (d) Output $\mathsf{sk}_{\mathsf{id}} = (\mathsf{sk}_{\mathsf{id}}^1, \ldots, \mathsf{sk}_{\mathsf{id}}^k)$.
3. $\mathsf{Enc}(\mathsf{pk}, m_1, \ldots, m_k)$. Upon input the public key pk and k plaintext messages $(m_1, \ldots, m_k) \in \mathbb{Z}_p^k$, proceed as follows:
   (a) Let $\mathbf{y} = (m_1, \ldots, m_k, 0, \ldots, 0) \in \mathbb{Z}_p^\ell$.
   (b) Output $C \leftarrow \mathcal{FE}.\mathsf{Enc}(\mathsf{pk}, \mathbf{y})$.
4. $\mathsf{Dec}(\mathsf{sk}_{\mathsf{id}}, C)$. Upon input the secret key $\mathsf{sk}_{\mathsf{id}}$ for user id and a ciphertext C, proceed as follows:
   (a) Parse $\mathsf{sk}_{\mathsf{id}} = (\mathsf{sk}_{\mathsf{id}}^1, \ldots, \mathsf{sk}_{\mathsf{id}}^k)$.
   (b) For each $j \in [k]$, compute $\tilde{m}_j = \mathcal{FE}.\mathsf{Dec}(\mathsf{pk}, C, \mathsf{sk}_{\mathsf{id}}^j)$.

   (c) Output $\tilde{\mathbf{m}} = (\tilde{m}_1, \dots, \tilde{m}_k)$.

5. $\mathsf{Trace}(\mathsf{pd}, \mathcal{S}, \mathcal{O}^{\mathcal{D}})$. Upon input the public directory pd, a suspect set $\mathcal{S}$ of users, and given access to the oracle $\mathcal{O}^{\mathcal{D}}$, execute Steps (a) through (e) for each component $j \in [k]$.

   First do the preprocessing for component j in step (a):

   (a) Uniformly sample $m_1, \dots, m_{j-1}, m_{j+1}, \dots, m_k \hookleftarrow \mathbb{Z}_p$ and find $m_j, m'_j \in \mathbb{Z}_p$ such that

   $$\left| \Pr_{C \hookleftarrow \mathsf{Enc}(\mathsf{pk}, \mathbf{y})} \left[{\mathcal{O}^{\mathcal{D}}}_j(m_j, C) = 1 \right] - \Pr_{C' \hookleftarrow \mathsf{Enc}(\mathsf{pk}, \mathbf{y}')} \left[{\mathcal{O}^{\mathcal{D}}}_j(m_j, C') = 1 \right] \right|$$

   is non-negligible, where we set $\mathbf{y} = (m_1, \dots, m_j, \dots, m_k, 0, \dots, 0)$ and $\mathbf{y}' = (m_1, \dots, m'_j, \dots, m_k, 0, \dots, 0)$.

   Set $\mathcal{S}_1 = \mathcal{S}$ and execute Steps (b) to (e) for the component j iteratively with $i = 1, 2, \dots$.

   (b) If $i = 1$, set $\mathbf{v}_{\mathcal{S}_i} = \mathbf{0}$. If $\mathcal{S}_i = \emptyset$, set $\mathbf{v}_{\mathcal{S}_i} = (0, \dots, 0, m'_j - m_j, 0, \dots, 0) \in \mathbb{Z}_p^\ell$. Else, compute $\mathbf{v}_{\mathcal{S}_i} \in \mathbb{Z}_p^\ell$ such that:[9]
      i. For all $\mathsf{id} \in \mathcal{S}_i$, for all $j_1 \in [k]$: $\langle \hat{\mathbf{x}}_{\mathsf{id}, j_1}, \mathbf{v}_{\mathcal{S}_i} \rangle = 0$.
      ii. For all $\mathsf{id} \in \mathcal{S} \setminus \mathcal{S}_i$:
         A. for all $j_1 \in [k] \setminus \{j\}$, $\langle \hat{\mathbf{x}}_{\mathsf{id}, j_1}, \mathbf{v}_{\mathcal{S}_i} \rangle = 0$,
         B. for $j_1 = j$, $\langle \hat{\mathbf{x}}_{\mathsf{id}, j_1}, \mathbf{v}_{\mathcal{S}_i} \rangle = (m'_j - m_j)$.

   (c) Repeat the following steps sufficiently many times (as dictated by Hoeffding's inequality) to compute an approximation of the probability p_i that the response from ${\mathcal{O}^{\mathcal{D}}}_j$ is $b_i = 1$.
      i. Construct $\mathbf{y} = \mathbf{v}_{\mathcal{S}_i} + (m_1, \dots, m_j, \dots, m_k, 0, \dots, 0) \in \mathbb{Z}_p^\ell$;
      ii. The probe ciphertext is $C^{\mathcal{S}_i} \leftarrow \mathcal{FE}.\mathsf{Enc}(\mathsf{pk}, \mathbf{y})$;
      iii. Provide the oracle ${\mathcal{O}^{\mathcal{D}}}_j$ with $(m_j, C^{\mathcal{S}_i})$ as input and get a binary value b_i as output.

   (d) If $i > 1$ and $|p_i - p_{i-1}|$ is non-negligible, then output id_{i-1} and abort;

   (e) If $\mathcal{S}_i = \emptyset$, then
      i. If $j = k$, then output $\perp$ and abort;
      ii. Else continue from Step (a) with $j = j + 1$;

   Else, set $\mathcal{S}_{i+1} = \mathcal{S}_i \setminus \{\mathsf{id}_i\}$.

We note here that the description of the Trace algorithm is such that it stops after finding one traitor for the first component $j \in [k]$ in the increasing order. This is in accordance with the definition of a tracing algorithm as in Definition 1. As in the $\mathcal{TR}$ scheme of Section 3.1, vector $\mathbf{v}_{\mathcal{S}_i}$ is computed using a deterministic algorithm.

We now check the correctness of the scheme.

Theorem 21. *Assume that $p = \lambda^{\omega(1)}$. Then, for every $j \in [k]$ and every $m_j \in \mathbb{Z}_p$, we have*

$$\mathsf{Dec}(\mathsf{sk}_{\mathsf{id}}^j, \mathsf{Enc}(\mathsf{pk}, m_1, \dots, m_k)) = m_j,$$

with probability $\geq 1 - \lambda^{-\omega(1)}$.

Proof. In the execution of $\mathsf{Dec}(\mathsf{sk}_{\mathsf{id}}, \mathsf{Enc}(\mathsf{pk}, m_1, \dots, m_k))$ for every component $\mathsf{sk}_{\mathsf{id}}^j$ of the secret key, we have:

[9] The set of linear equations has rank $r \leq k + t - 1 < \ell$. Hence there exists a solution vector $\mathbf{v}_{\mathcal{S}_i}$.

$$\begin{aligned}\mathsf{Dec}(\mathsf{sk}_{\mathsf{id}}^j, C) &= \mathcal{FE}.\mathsf{Dec}(\mathsf{pk}, C, \mathsf{sk}_{\mathsf{id}}^j)\\ &= \langle \hat{\mathbf{x}}_{\mathsf{id},j}, (m_1, \ldots, m_k, 0, \ldots, 0)\rangle\\ &= \langle (\mathbf{1}_j, \mathbf{x}_{\mathsf{id}}), (m_1, \ldots, m_k, 0, \ldots, 0)\rangle\\ &= \langle (\mathbf{1}_j, (m_1, \ldots, m_k)\rangle + \langle \mathbf{x}_{\mathsf{id}}, (0, \ldots, 0)\rangle\\ &= m_j,\end{aligned}$$

by correctness of $\mathcal{FE}$. □

Like in the $\mathcal{TR}$ scheme, there are several choices for m_j and m'_j for the Trace algorithm of the $\mathcal{TT}$ scheme. The proof is similar to that of Lemma 14.

Lemma 22. *Let the decoder $\mathcal{D}$ output by the adversary $\mathcal{A}$ in the game* AD-TT *be useful for component j as defined in Equation* (1)*. Then, with probability $\geq 1/(4\lambda^{c_j})$ over random choices of $m_1, \ldots, m_{j-1}, m_j, m'_j, m_{j+1}, \ldots, m_k \hookleftarrow \mathbb{Z}_p$, we have:*

$$\left| \Pr_{C \hookleftarrow \mathsf{Enc}(\mathsf{pk}, \mathbf{y})} \left[\mathcal{O}^{\mathcal{D}}{}_j(m_j, C) = 1 \right] - \Pr_{C' \hookleftarrow \mathsf{Enc}(\mathsf{pk}, \mathbf{y}')} \left[\mathcal{O}^{\mathcal{D}}{}_j(m_j, C') = 1 \right] \right| \geq \frac{1}{2\lambda^{c_j}}$$

where $\mathbf{y} = (m_1, \ldots, m_j, \ldots, m_k, 0, \ldots, 0)$ and $\mathbf{y}' = (m_1, \ldots, m'_j, \ldots, m_k, 0, \ldots, 0)$.

Based on Lemma 22, the preprocessing in Step (a) of Algorithm Trace can be implemented efficiently by repeatedly sampling $m_j, m'_j \hookleftarrow \mathcal{M}$ and estimating the probabilities that $\mathcal{O}^{\mathcal{D}}{}_j(m_j, C) = 1$ and $\mathcal{O}^{\mathcal{D}}{}_j(m, C') = 1$ using Hoeffding's bound, until the probability difference is sufficiently large.

B.2 Semantic Security

In the following, we prove IND-CPA security of our scheme.

Theorem 23. *If $\mathcal{FE}$ is* IND-CPA *secure, then $\mathcal{TT}$ is* IND-CPA *secure as well.*

Proof. Let $\mathcal{A}_{\mathcal{TT}}$ be a probabilistic polynomial-time adversary that breaks semantic security of $\mathcal{TT}$. We construct a probabilistic polynomial-time adversary $\mathcal{A}_{\mathcal{FE}}$ that breaks semantic security of $\mathcal{FE}$. Adversary $\mathcal{A}_{\mathcal{FE}}$ proceeds as follows.

- It first obtains the public key pk output by the $\mathcal{FE}$ challenger (who runs the $\mathcal{FE}.\mathsf{Setup}(1^\lambda, 1^\ell)$ algorithm) and relays it to $\mathcal{A}_{\mathcal{TT}}$.
- When $\mathcal{A}_{\mathcal{TT}}$ chooses two message vectors

  $$(m_1^0, \ldots, m_k^0), (m_1^1, \ldots, m_k^1) \in \mathbb{Z}_p^k$$

  and provides them to $\mathcal{A}_{\mathcal{FE}}$, adversary $\mathcal{A}_{\mathcal{FE}}$ proceeds as follows:
  - It sends $\mathbf{y}_0 = (m_1^0, \ldots, m_k^0, 0, \ldots, 0)$ and $\mathbf{y}_1 = (m_1^1, \ldots, m_k^1, 0, \ldots, 0)$ to the $\mathcal{FE}$ challenger who samples $b \hookleftarrow \{0, 1\}$ and encrypts $\mathbf{y}_b$ as $C_{\mathbf{y}_b} \leftarrow \mathcal{FE}.\mathsf{Enc}(\mathsf{pk}, \mathbf{y}_b)$.
  - Adversary $\mathcal{A}_{\mathcal{FE}}$ receives $C_{\mathbf{y}_b}$ from the $\mathcal{FE}$ challenger and relays it to $\mathcal{A}_{\mathcal{TT}}$.
- Finally, adversary $\mathcal{A}_{\mathcal{TT}}$ outputs its guess $b' \in \{0, 1\}$ and $\mathcal{A}_{\mathcal{FE}}$ also outputs b' as its own guess of b.

Note that adversary $\mathcal{A}_{\mathcal{FE}}$ behaves exactly as an IND-CPA challenger in the view of $\mathcal{A}_{\mathcal{TT}}$ and is also a valid IND-CPA adversary against the $\mathcal{FE}$ challenger. The advantage of $\mathcal{A}_{\mathcal{FE}}$ is exactly the same as the advantage of $\mathcal{A}_{\mathcal{TT}}$. □

Note that the semantic security of the $\mathcal{TT}$ scheme is independent of the bound on the number of colluders in the $\mathcal{FE}$ scheme. The $\mathcal{FE}$ collusion bound only plays a role in the bound on the number of traitors for the traceability proofs.

B.3 Traceability

Since the $\mathcal{TT}$ scheme above is very similar to the $\mathcal{TR}$ scheme of Section 3.1, all the proofs go through with changes required for considering the message and the secret key as vectors and the modified definition of usefulness. As mentioned before, the bound r on the number of revoked users is no more relevant in the context of the $\mathcal{TT}$ scheme. However, we have an additional parameter k for the number of messages encrypted into one ciphertext. Consequently, in order to deal with t traitors, we need to work with vectors of dimension $\ell = k + t$.

Now, in the security games for $\mathcal{TR}$, when the $\mathsf{AD\text{-}TT}$ adversary $\mathcal{A}$ queries the $\mathcal{FE}$ adversary with an $\mathbf{x}_{\mathsf{id}}$ for an id that it chooses adaptively, it is relayed to the $\mathcal{FE}$ challenger. Here, for the $\mathcal{TT}$ scheme, the $\mathbf{x}_{\mathsf{id}}$ is queried by $\mathcal{A}$ as before. However, the adversary $\mathcal{A}_{\mathcal{FE}}$ expands that $\mathbf{x}_{\mathsf{id}}$ to k many vectors of length ℓ, namely $\hat{\mathbf{x}}_{\mathsf{id},j}$ for each $j \in [k]$. When the $\mathcal{FE}$ challenger returns the $\mathsf{sk}^j_{\mathsf{id}}$ for each $j \in [k]$, the secret key vector $\mathsf{sk}_{\mathsf{id}} = (\mathsf{sk}^1_{\mathsf{id}}, \ldots, \mathsf{sk}^k_{\mathsf{id}})$ is relayed to $\mathcal{A}$. Consequently, the adversary $\mathcal{A}_{\mathcal{FE}}$ makes $k \cdot t$ key queries to the $\mathcal{FE}$ challenger.

We assume that the decoder $\mathcal{D}$ produced by the adversary $\mathcal{A}$ in the Trace algorithm, is useful for at least one component $j \in [k]$ of the message vector, as indicated in Equation (5). The Trace algorithm has access to this decoder through the oracle $\mathcal{O}^{\mathcal{D}}$ that outputs a k-tuple $\mathbf{b}$ of bits. Oracle ${\mathcal{O}^{\mathcal{D}}}_j$ effectively extracts the jth bit b_j from $\mathbf{b}$. Let $\mathcal{T}$ (respectively $\mathcal{S}$) be the set of identities of traitors (respectively traitor suspects) that have leaked various components of their secret key vectors. As in the $\mathcal{TR}$ scheme, we assume that $\mathcal{T} \subseteq \mathcal{S}$, as specified in game $\mathsf{AD\text{-}TT}$. Without loss of generality, we may assume $|\mathcal{S}| = t$. We show in this section that $\mathsf{Trace}(\mathsf{pd}, \mathcal{S}, \mathcal{O}^{\mathcal{D}})$ outputs an identity $\mathsf{id} \in \mathcal{T}$.

Lemma 24. *If $\mathcal{FE}$ is $\mathsf{IND\text{-}CPA}$ secure, then $\mathcal{TR}$ satisfies public traceability.*

Proof. Let us assume by contradiction that an adversary $\mathcal{A}$ can break the public traceability of $\mathcal{TR}$ with non-negligible probability. We then construct a probabilistic polynomial-time adversary $\mathcal{A}_{\mathcal{FE}}$ that breaks the semantic security of $\mathcal{FE}$. Adversary $\mathcal{A}_{\mathcal{FE}}$ proceeds as follows.

- It first obtains the public key pk output by the $\mathcal{FE}$ challenger (who runs the $\mathcal{FE}.\mathsf{Setup}(1^\lambda, 1^\ell)$ algorithm) and relays it to the adversary $\mathcal{A}$.
- When $\mathcal{A}$ asks $\mathcal{A}_{\mathcal{FE}}$ to create a p_{id} for some id, $\mathcal{A}_{\mathcal{FE}}$ in turn asks the $\mathcal{FE}$ challenger to do the same. The $\mathcal{FE}$ challenger randomly chooses a vector $\mathbf{x}_{\mathsf{id}} \hookleftarrow \mathbb{Z}_p^\ell$ and sends it to $\mathcal{A}_{\mathcal{FE}}$ who further relays it to $\mathcal{A}$.
- When $\mathcal{A}$ makes a key query for an identity id it is added to the (initially empty) set $\mathcal{T}$ and $\mathcal{A}_{\mathcal{FE}}$ extends $\mathbf{x}_{\mathsf{id}}$ to get k vectors $\hat{\mathbf{x}}_{\mathsf{id},j} = (\mathbf{1}_j, \mathbf{x}_{\mathsf{id}}), j \in [k]$ each of dimension $\ell = k + t$. It then queries the $\mathcal{FE}$ challenger for secret keys for each $\hat{\mathbf{x}}_{\mathsf{id},j}, j \in [k]$ and receives the corresponding key $\mathsf{sk}^j_{\mathsf{id}}$; it constructs the key vector $\mathsf{sk}_{\mathsf{id}} = (\mathsf{sk}^1_{\mathsf{id}}, \ldots, \mathsf{sk}^k_{\mathsf{id}})$ of keys and sends it to $\mathcal{A}$.[10]

Note that $\mathcal{A}$ makes at most t key queries, each of which are translated by $\mathcal{A}_{\mathcal{FE}}$ into k key queries to the $\mathcal{FE}$ challenger.

[10] Although we assume that all key components of an $\mathsf{sk}_{\mathsf{id}}$ may not be leaked, by allowing the adversary $\mathcal{A}_{\mathcal{FE}}$ and in turn $\mathcal{A}$ to have all the components of $\mathsf{sk}_{\mathsf{id}}$, we give it more power. As a result, if $\mathcal{A}$ uses all the component keys to construct $\mathcal{D}$, decoder $\mathcal{D}$ may be useful for all components $j \in [k]$. We note here that the decryption capability for a component depends on the secret key of only that component.

- Adversary $\mathcal{A}$ finally produces a pirate decoder $\mathcal{D}$ and chooses a suspect set $\mathcal{S}$ of cardinality $\leq t$ that contains $\mathcal{T}$.[11] Then, the adversary $\mathcal{A}_{\mathcal{FE}}$ executes the Trace algorithm on $\mathcal{O}^{\mathcal{D}}$ to find j and i such that for the component j, the quantity $|p_i - p_{i-1}|$ is non-negligible. If Trace aborts with output $\perp$ or index i such that $\mathsf{id}_{i-1} \notin \mathcal{T}$ but $\mathsf{id}_{i-1} \in \mathcal{S}$, $\mathcal{A}_{\mathcal{FE}}$ also aborts and outputs a random bit. Otherwise, it sets
$$\mathbf{y}_0 = \mathbf{v}_{\mathcal{S}_{i-1}} + (m_1, \ldots, m_j, \ldots, m_k, 0, \ldots, 0) \text{ and}$$
$$\mathbf{y}_1 = \mathbf{v}_{\mathcal{S}_i} + (m_1, \ldots, m'_j, \ldots, m_k, 0, \ldots, 0)$$
and sends them as challenge messages to the $\mathcal{FE}$ challenger.[12]
- The $\mathcal{FE}$ challenger samples a bit $b \hookleftarrow \{0,1\}$ and sends the ciphertext $\mathcal{FE}.\mathsf{Enc}(\mathsf{pk}, \mathbf{y}_b)$ to $\mathcal{A}_{\mathcal{FE}}$. The adversary $\mathcal{A}_{\mathcal{FE}}$ runs $\mathcal{O}^{\mathcal{D}}{}_j$ on input (m_j, C_b), where $C_b = \mathcal{FE}.\mathsf{Enc}(\mathsf{pk}, \mathbf{y}_b)$. Then $\mathcal{O}^{\mathcal{D}}{}_j$ outputs the bit $b' \in \{0,1\}$.
- Finally, adversary $\mathcal{A}_{\mathcal{FE}}$ outputs the same bit $b' \in \{0,1\}$ if $p_i - p_{i-1} > 0$ and $1 - b'$ otherwise.

In the above game, we note that the sets $\mathcal{T}$ and $\mathcal{S}$ were chosen by $\mathcal{A}$ such that $\mathsf{id}_{i-1} \notin \mathcal{T}$ but $\mathsf{id}_{i-1} \in \mathcal{S}$. That $\mathcal{A}_{\mathcal{FE}}$ is a valid $\mathcal{TT}$ challenger, is quite clear. We argue that $\mathcal{A}_{\mathcal{FE}}$ is a valid adversary against the $\mathcal{FE}$ challenger. Note that when $\mathcal{A}_{\mathcal{FE}}$ does not abort in Step (d), we have $\mathsf{id}_{i-1} \notin \mathcal{T}$ but $\mathsf{id}_{i-1} \in \mathcal{S}$. The keys queried by $\mathcal{A}_{\mathcal{FE}}$ are for $\mathsf{id} \in \mathcal{T}$ only. This set $\mathcal{T}$ can be partitioned into $\mathcal{T} \cap \mathcal{S}_{i-1}$ and $\mathcal{T} \cap (\mathcal{S}_1 \setminus \mathcal{S}_{i-1})$. We show that decryptions of $\mathcal{FE}.\mathsf{Enc}(\mathsf{pk}, \mathbf{y}_0)$ and $\mathcal{FE}.\mathsf{Enc}(\mathsf{pk}, \mathbf{y}_1)$ result in the same value for all keys queried by $\mathcal{A}_{\mathcal{FE}}$.

1. For $\mathsf{id} \in \mathcal{S}_{i-1}$, we have
   - $\langle \hat{\mathbf{x}}_{\mathsf{id},j_1}, \mathbf{y}_0 \rangle = \langle \hat{\mathbf{x}}_{\mathsf{id},j_1}, \mathbf{y}_1 \rangle = m_{j_1}$ for all $j_1 \in [k]$.

   Hence for all $\mathsf{id} \in \mathcal{T} \cap \mathcal{S}_{i-1}$ for which the $\mathsf{sk_{id}}$ was queried by $\mathcal{A}_{\mathcal{FE}}$, the inner products $\langle \hat{\mathbf{x}}_{\mathsf{id},j_1}, \mathbf{y}_0 \rangle$ and $\langle \hat{\mathbf{x}}_{\mathsf{id},j_1}, \mathbf{y}_1 \rangle$ have the same value for all $j_1 \in [k]$.
2. Similarly, for $\mathsf{id} \in \mathcal{T} \cap (\mathcal{S}_1 \setminus \mathcal{S}_{i-1})$, we have
   - $\langle \hat{\mathbf{x}}_{\mathsf{id},j_1}, \mathbf{y}_0 \rangle = \langle \hat{\mathbf{x}}_{\mathsf{id},j_1}, \mathbf{y}_1 \rangle = m'_j$, for $j_1 = j$, and
   - $\langle \hat{\mathbf{x}}_{\mathsf{id},j_1}, \mathbf{y}_0 \rangle = \langle \hat{\mathbf{x}}_{\mathsf{id},j_1}, \mathbf{y}_1 \rangle = m_{j_1}$, for all $j_1 \in [k] \setminus \{j\}$.

   Hence for all $\mathsf{id} \in \mathcal{T} \cap (\mathcal{S}_1 \setminus \mathcal{S}_{i-1})$ for which the $\mathsf{sk_{id}}$ was queried by $\mathcal{A}_{\mathcal{FE}}$, the inner products $\langle \hat{\mathbf{x}}_{\mathsf{id},j_1}, \mathbf{y}_0 \rangle$ and $\langle \hat{\mathbf{x}}_{\mathsf{id},j_1}, \mathbf{y}_1 \rangle$ have the same value for all $j_1 \in [k]$.

Hence, $\mathcal{A}_{\mathcal{FE}}$ is a valid adversary against the $\mathcal{FE}$ challenger.

Another point to be noted in the above game is that each key query for an $\mathsf{id} \in \mathcal{T}$ by $\mathcal{A}$ is transformed into k $\mathcal{FE}$ key queries by $\mathcal{A}_{\mathcal{FE}}$. For each $\mathsf{id} \in \mathcal{T}$, the adversary $\mathcal{A}_{\mathcal{FE}}$ makes k key queries to the $\mathcal{FE}$ challenger one for each $\hat{\mathbf{x}}_{\mathsf{id},j}, j \in [k]$. Each $\mathsf{sk_{id}}$ received by $\mathcal{A}$ is a k-tuple. Hence in total, the $k \cdot t$ keys for $\mathcal{FE}$ received by $\mathcal{A}_{\mathcal{FE}}$ translate into the t keys [13] for $\mathcal{TT}$ received by $\mathcal{A}$ where each $\mathsf{sk_{id}}$ it receives is a k-tuple. Hence, $\mathcal{A}_{\mathcal{FE}}$ is a valid adversary against the $\mathcal{FE}$ challenger and a valid $\mathcal{TT}$ challenger as well.

We recollect that in the $\mathsf{AD\text{-}TT}$ game, we say that $\mathcal{A}$ wins if the decryption box $\mathcal{D}$ output by it is such that when Trace is executed on input $\mathcal{O}^{\mathcal{D}}$, it fails to identify a traitor. In such a case, Trace either outputs $\perp$ or it outputs an $\mathsf{id}_{i-1} \notin \mathcal{T}$ with probability at least $1/\lambda^c$. We next argue that if $\mathcal{A}$ outputs $\mathcal{D}$ that satisfies this winning

[11] Recall that we assume that $\mathcal{D}$ is stateless/resettable and replies independently to successive queries.

[12] Here, $m_1, \ldots, m_j, m'_j, \ldots, m_k$ are chosen as explained by Lemma 22, and $\mathbf{v}_{\mathcal{S}_{i-1}}, \mathbf{v}_{\mathcal{S}_i} \in \mathbb{Z}_p^\ell$ are chosen as in Step (b) of algorithm Trace.

[13] Note that the $k \cdot t$ queries made by $\mathcal{A}_{\mathcal{FE}}$ are not uniform and independent. In fact, the $k \cdot t$ keys received by $\mathcal{A}_{\mathcal{FE}}$ span a space of dimension $k + t - 1$. This is why we require more than q-$\mathsf{IND\text{-}CPA}$ security from $\mathcal{FE}$.

condition of the $\mathsf{AD\text{-}TT}$ game, then $\mathcal{A}_{\mathcal{FE}}$ has non-negligible advantage in the above game. To see this, we first observe that when $\mathcal{A}_{\mathcal{FE}}$ returns a random bit, it correctly guesses b with probability $1/2$ since its output is random in this case. Then, it suffices to show the following:

- In the above game, adversary $\mathcal{A}_{\mathcal{FE}}$ does not abort with non-negligible probability.
- Conditioned on $\mathcal{A}_{\mathcal{FE}}$ not aborting, the quantity $|p_i - p_{i-1}|$ is non-negligible.

The second item follows because if $|p_i - p_{i-1}|$ is not sufficiently large, then $\mathcal{A}_{\mathcal{FE}}$ does not output id_{i-1} at Step (d) of Trace except for a negligible probability because of the Hoeffding bound.

Next, we prove the first item. Let us assume that $\mathcal{A}$ outputs $\mathcal{D}$ that satisfies the winning condition. This happens with non-negligible probability by our hypothesis. Therefore, it suffices to show that $\mathcal{A}$ does not abort with non-negligible probability conditioned on the event. Since we are assuming $\mathcal{D}$ satisfies the winning condition, when Trace is executed on input $\mathcal{O}^{\mathcal{D}}{}_j$ for all $j \in [k]$, we have that for some $j \in [k]$, oracle $\mathcal{O}^{\mathcal{D}}{}_j$ outputs $\mathsf{id}_{i-1} \notin \mathcal{T}$ or outputs $\perp$ and aborts with probability at least $1/\lambda^c$. On the other hand, we observe that $\mathcal{A}_{\mathcal{FE}}$ does not abort only when Trace on input $\mathcal{O}^{\mathcal{D}}$ outputs $\mathsf{id}_{i-1} \notin \mathcal{T}$. Therefore, to conclude the proof, it suffices to show the following lemma, which says that the probability of Trace giving output $\perp$ and aborting is negligible. □

Theorem 25. *Assume that a pirate decoder $\mathcal{D}$ satisfies Equation* (1)*. Then, the execution of* Trace *returns some* $\mathsf{id} \in \mathcal{S}$ *(and not $\perp$) with overwhelming probability.*

Proof. We consider a variant of Trace that continues its execution until it exhausts $\mathcal{S}$ for each $j \in [k]$, even if it has already output an id.

Note that for a component $j \in [k]$ for which the decryption box $\mathcal{D}$ is not useful, the probability

$$\Pr_{\substack{\mathbf{m} \leftarrow U(\mathcal{M}^k) \\ C \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathbf{m})}} \left[\mathcal{O}^{\mathcal{D}}{}_j(m_j, C) = 1\right]$$

in Equation (5) that is captured by p_1 will be negligibly close to $\frac{1}{|\mathcal{M}|}$. In the iterations of steps (b) to (e) of Trace, as the value of i increases,

- for all $i > 1$, each p_i will be negligibly close to $\frac{1}{|\mathcal{M}|}$ and hence $|p_i - p_{i-1}|$ will be negligible as well;
- as a consequence, the set $\mathcal{S}_i$ gets smaller until it is finally empty and Trace moves on to investigate the next component $j+1$.

After possible iterations of components that are not useful, Trace lands at a component j for which $\mathcal{D}$ is useful. We consider the probabilities p_i at the start and end of the modified execution for a component $j \in [k]$ for which $\mathcal{D}$ is useful.

1. At the beginning, Trace considers $\mathcal{S}_1 = \mathcal{S}$ and $\mathbf{v}_{\mathcal{S}_1} = \mathbf{0}$. Hence, the genuine ciphertext output by the Enc algorithm and the probe ciphertext created by the Trace algorithm for the suspect subset $\mathcal{S}_1$ are exactly the same.
2. When $i = |\mathcal{S}| + 1$, we have $\mathcal{S}_i = \emptyset$ and $\mathbf{v}_{\mathcal{S}_i} = (0, \ldots, 0, m'_j - m_j, 0, \ldots, 0)$. In Step (b) of the Trace algorithm, the messages m_j and m'_j were chosen such that the difference in the probabilities p_1 and $p_{|\mathcal{S}|+1}$ is $\geq 1/(2\lambda^c)$.

Note that the two latter observations imply, via the triangle inequality, that for the component $j \in [k]$ for which $\mathcal{D}$ is useful, there exists an i such that $|p_i - p_{i-1}|$ is non-negligible. By the Hoeffding bound, Trace algorithm outputs id_{i-1} with overwhelming probability. □

B.4 A Traitor Tracing Scheme based on DDH

Abdalla *et al.* proposed in [1] the first direct construction of an IPFE, along with a proof of selective security holding under the DDH hardness assumption. Later, Agrawal *et al.* and Abdalla *et al.* [4, 2] proposed variants and proved adaptively secure, also under DDH.

These three schemes are not IPFEs over $\mathbb{Z}_p^\ell$ as defined in Subsection 2.2. Indeed, the adversary is allowed to query secret keys corresponding to arbitrary vectors in $\mathbb{Z}_p^\ell$, but the decryption proceeds only if the evaluated inner product belongs to a small set of values (for example, if the evaluated inner product is small) as it involves the computation of a discrete logarithm. From the functionality perspective, our generic transformation from an IPFE over $\mathbb{Z}_p^\ell$ to a traitor tracing scheme works if we impose a limitation that the plaintexts are small-norm vectors. The multi-messages version allows us to significantly enlarge the plaintext space.

Below, we describe a concrete DDH-based traitor tracing scheme for k messages, based on the IPFE from [4].

- $\mathsf{Setup}(1^\lambda, 1^k, 1^t)$. Choose a cyclic group $\mathbb{G}$ of prime order $p > 2^\lambda$ with generators $g, h \hookleftarrow \mathbb{G}$. Set $\ell = k + t$ and sample $\mathbf{s}, \mathbf{t} \hookleftarrow \mathbb{Z}_p^\ell$ and compute $\{h_i = g^{s_i} \cdot h^{t_i}\}_{i \in [\ell]}$. Define $\mathsf{msk} := (\mathbf{s}, \mathbf{t})$ and $\mathsf{pk} := (t, \mathbb{G}, g, h, p, \{h_i\}_{i \in [\ell]})$.
- $\mathsf{Keygen}(\mathsf{msk}, \mathsf{id})$. Sample $\mathbf{x}_{\mathsf{id}} \hookleftarrow \mathbb{Z}_p^t$. Then, for each $j \leq k$:
  - Define $\mathbf{x}_{u,j}$ as the concatenation between the jth canonical vector of dimension k and $\mathbf{x}_{\mathsf{id}}$.
  - Compute $\mathsf{sk}_{\mathsf{id},j} = (s_{\mathbf{x}_{\mathsf{id},j}}, t_{\mathbf{x}_{\mathsf{id},j}}) = (\langle \mathbf{s}, \mathbf{x}_{\mathsf{id},j} \rangle, \langle \mathbf{t}, \mathbf{x}_{\mathsf{id},j} \rangle)$.

  Set $\mathsf{sk}_{\mathsf{id}} = (\mathsf{sk}_{\mathsf{id},j})_{j \in [k]}$ and $p_{\mathsf{id}} = \mathbf{x}_{\mathsf{id}}$.
- $\mathsf{Enc}(\mathsf{pk}, \mathbf{m})$. Set $\mathbf{y} \in \mathbb{Z}_p^\ell$ as the concatenation between $\mathbf{m} \in \{0,1\}^k$ and $\mathbf{0}^t$. Sample $r \hookleftarrow \mathbb{Z}_p$. Compute $C = g^r, D = h^r$, and $\{E_i = g^{y_i} h_i^r\}_{i \in [\ell]}$. Return ciphertext $C_{\mathbf{m}} = (C, D, E_1, \ldots, E_\ell)$.
- $\mathsf{Dec}(\mathsf{sk}_{\mathsf{id}}, C_{\mathbf{m}})$. Given $\{\mathsf{sk}_{\mathsf{id},j} = (s_{\mathbf{x}_{\mathsf{id},j}}, t_{\mathbf{x}_{\mathsf{id},j}})\}_{j \in [k]}$ and $C = (C, D, E_0, \ldots, E_\ell)$, compute:

  $$E_{\mathbf{x}_{\mathsf{id},j}} = \Big(\prod_{i=1}^{\ell} E_i^{x_{\mathsf{id},j,i}} \Big) / \Big(C^{s_{\mathbf{x}_{\mathsf{id},j}}} \cdot D^{t_{\mathbf{x}_{\mathsf{id},j}}} \Big).$$

  Then compute and return $\{\log_g (E_{\mathbf{x}_{\mathsf{id},j}})\}_{j \in [k]}$.
- $\mathsf{Trace}(\mathsf{pd}, \mathcal{S}, \mathcal{O}^{\mathcal{D}})$ proceeds as described in Section B.1. In particular, in the iterations of Steps (b) to (e), for a suspect set $\mathcal{S}_i$ of size $\leq t$, a probe ciphertext is created as follows:
  1. Find $m_1, \ldots, m_j, m'_j, \ldots, m_k$, as described in Step (a).
  2. Compute $\mathbf{v}_{\mathcal{S}_i} \in \mathbb{Z}_p^\ell$ as described in Step (b).
  3. Set $\mathbf{y} = \mathbf{v}_{\mathcal{S}_i} + (m_1, \ldots, m_j, \ldots, m_k, 0, \ldots, 0)$.
  4. Sample $r \hookleftarrow \mathbb{Z}_p$ and compute $C = g^r, D = h^r, \{E_i = g^{y_i} h_i^r\}_{i=1}^{\ell}$ where y_i denotes the ith component of the vector $\mathbf{y}$.

  The probe ciphertext is $C_{\mathbf{m}} = (C, D, E_1, \ldots, E_\ell)$.

The public key contains $k + t + 2$ group elements, the master secret key consists of $2(k + t)$ elements of $\mathbb{Z}_p$, and to every user corresponds a secret key of $2k$ elements of $\mathbb{Z}_p$ and a vector of $k+t$ group elements. Enc maps a plaintext of k bits to a ciphertext of $k + t + 2$ group elements.

Note that decryption is efficient as for any valid ciphertext and for any j, the quantity $E_{\mathbf{x}_{id,j}}$ can take at most 2 values. It is possible to decrease k, and hence ciphertext expansion and the number of secret key vectors, by increasing the set of possible values for each plaintext component m_j. This impacts the decryption cost.