Citation for published version


DOI

https://doi.org/10.1145/3133956.3134041

Link to record in KAR

https://kar.kent.ac.uk/83284/

Document Version

Author's Accepted Manuscript
Efficient Public Trace and Revoke from Standard Assumptions

Shweta Agrawal
IIT Madras, India,
shweta@iitm.ac.in
and
Sanjay Bhattacherjee
Turing Lab (ASU), ISI Kolkata, India,
sanjay.bhattacherjee@gmail.com
and
Duong Hieu Phan
XLIM (U. Limoges, CNRS), France,
duong-hieu.phan@unilim.fr
and
Damien Stehlé
ENS de Lyon, LIP (U. Lyon, CNRS, ENSL, INRIA, UCBL), France,
damien.stehle@ens-lyon.fr
and
Shota Yamada
National Institute of Advanced Industrial Science and Technology (AIST), Japan,
yamada-shota@aist.go.jp

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 Composite Residuosity.

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

1 Introduction

A traitor tracing system Chor et al. (1994) 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 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 Fiat and Naor (1993), 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 Naor and Pinkas (2000); Naor et al. (2001); Dodis and Fazio (2003); Phan and Trinh (2011) and are notoriously hard to construct (please see Boneh and Waters (2006) for a detailed discussion). A desirable attribute for trace-and-revoke systems is public traceability, meaning 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, trace-and-revoke systems with public traceability have only been achieved by Boneh and Waters Boneh and Waters (2006), and quite recently by Nishimaki, Wichs and Zhandry (NWZ) Nishimaki et al. (2016). 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 $\lambda$. 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) Nishimaki et al. (2016), all known schemes in this model Boneh and Franklin (1999); Hofheinz and Striecks (2014); Ling et al. (2014) support either revocation or public traceability but not both.

Recently, Nishimaki et al. (NWZ) Nishimaki et al. (2016) provided a generic construction for traitor tracing systems from functional encryption schemes. Functional encryption Sahai and Waters (2005); Boneh et al. (2011) is a generalization of public key encryption allowing fine grained access to encrypted data. We note that the strongest constructions in Nishimaki et al. (2016) are based on the existence of indistinguishability obfuscation Barak et al. (2012), 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 Gorbunov et al. (2012) or subexponential time hardness of learning with errors (LWE) Goldwasser et al. (2013); Agrawal and Rosen (2016). 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 Gorbunov et al. (2012), the resulting scheme has a ciphertext size growing at least as $O(r + t)^5 \text{Poly}(\lambda)$ where $r$ is the maximum size of the list of revoked users and $t$ the maximum coalition size (please see Appendix 6 for an explanation of the bound). By relying on learning with errors, this blowup can be improved to $O((r + t)^4 \text{Poly}(\lambda))$ but at the cost of relying on heavy machinery such as attribute based encryption Gorbunov et al. (2013) and fully homomorphic encryption Goldwasser et al. (2013). 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 Agrawal and Rosen (2016) leads to better asymptotic
bounds $O(r + t)^3 Poly(\lambda))$ 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 Nishimaki et al. (2016). 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) Abdalla et al. (2015); Agrawal et al. (2016) suffices. Efficient constructions for IPFE satisfying adaptive security are available Agrawal et al. (2016), leading to trace-and-revoke systems which are significantly simpler and more efficient than those implied by Nishimaki et al. (2016). 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 Composite Residuosity (DCR), which enjoy substantial benefits over using those of Agrawal et al. (2016) 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 with bounded collusion resistance, from standard assumptions. Our schemes support public, black-box traceability and achieve the strongest notion of adaptive security as defined by Boneh and Waters (2006). Our construction is generic and leverages recent constructions of modular inner product functional encryption (IPFE) Abdalla et al. (2015); Agrawal et al. (2016). 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 Nishimaki et al. (2016) 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 $\tilde{O}((r + t)\lambda^3)$, while our LWE-based construction has ciphertext size $\tilde{O}(r + t + \lambda)$ and key size $\tilde{O}((r + t + \lambda)\lambda)$. We note that our security definition considers the strongest notion of “usefulness” Boneh and Waters (2006) of the pirate decoder, which is not satisfied by most other constructions. Indeed some schemes Naor and Pinkas (2000); Dodis and Fazio (2003) are actually insecure in this strong game (see Appendix 6 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 $\mathcal{FE} = (\mathcal{FE}.\text{Setup}, \mathcal{FE}.\text{KeyGen}, \mathcal{FE}.\text{Enc}, \mathcal{FE}.\text{Dec})$ 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 $v \in \mathbb{Z}_p^\ell$, the secret key encodes a vector $x \in \mathbb{Z}_p^\ell$ and decryption recovers the inner product $\langle x, v \rangle \mod p$.

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 $x_{id} \in \mathbb{Z}_p^\ell$ and the entry $pd_{id} = (id, x_{id})$ 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.

\(^1\)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.
To revoke a set $R$ of users with $|R| \leq r$, we first deterministically compute a vector $v_R \in \mathbb{Z}_p^l$ such that for all $i \in R$, we have $\langle x_{id}, v_R \rangle = 0$ (modulo $p$). Note that this can be implemented only if $r < \ell$. At the same time, for a user $id \notin R$, the probability that $\langle x_{id}, v_R \rangle = 0$ 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 $x_{id}$ uniformly random, we have $\langle x_{id}, v_R \rangle \neq 0$ for $id \notin R$ with overwhelming probability.

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

$$C = (FE.Enc(pk, m \cdot v_R), R) = (C_1, C_2).$$

Here the operation $\cdot$ denotes the scalar multiplication of each component of $v_R$ with $m$. To decrypt, the user $id$ with the vector $x_{id}$ and the $FE$ secret key $sk_{x_{id}}$ proceeds as follows:

(a) Compute $v_R$ from $R$ and abort if $\langle x_{id}, v_R \rangle = 0$.

(b) If $\langle x_{id}, v_R \rangle \neq 0$, compute

$$\frac{FE.Dec(sk_{x_{id}}, C_1)}{\langle x_{id}, v_R \rangle} = \frac{\langle x_{id}, m \cdot v_R \rangle}{\langle x_{id}, v_R \rangle} = m.$$

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 $D$ and a set $S = \{id_1, id_2, \ldots, \}$ of suspected traitors with $|S| \leq t$, it is possible to find an identity $id$ in the set $T$ of traitors, as long as $T \subseteq S$. Here, we assume $R \cap S = \emptyset$ for simplicity.

Given a pirate decoder $D$, our tracing algorithm first finds a pair of messages $m$ and $m'$ such that $D$ 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 $S_i = \{id_i, id_{i+1}, \ldots \}$ for $i = 1, \ldots, |S| + 1$. We then generate a probe ciphertext $C^S_i$ associated to $S_i$ with the following properties:

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

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

We prove that the tracing algorithm always outputs some user in $T$. To see this, we first observe that by the first and second properties above, the decoder $D$ distinguishes $C^{S_1} = C^S$ and $C^{S_{|S|+1}} = C^\emptyset$ with non-negligible advantage. Therefore, by the triangle inequality, there exists at least one index $i$ such that $D$ distinguishes $C^{S_{i-1}}$ and $C^{S_i}$ 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 $v_S \in \mathbb{Z}_p^l$ as follows: If $i = 1$, we set $v_S = 0$; if $i = |S| + 1$, we set $v_S = (m' - m) \cdot v_R$ where $v_R$ is chosen as in the ordinary encryption algorithm; Otherwise, we set $v_S$ so that

- $\langle x_{id}, v_S \rangle = 0$ for every $id \in S_i \cup R$,
- $\langle x_{id}, v_S \rangle = (m' - m) \cdot \langle x_{id}, v_R \rangle$ for every $id \in S_i \setminus S_i$.  

4
Note that this can be implemented only if \( r + t < \ell \). We then set the probe ciphertext as follows:

\[
C_{S_i} = (C_1, C_2) = (\mathcal{FE}.\text{Enc}(pk, v_{S_i} + m \cdot v_{R}), R).
\]

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 \( sk_{x_{id}} \) for \( id \). We have

\[
\frac{\mathcal{FE}.\text{Dec}(sk_{x_{id}}, C_1)}{\langle x_{id}, v_R \rangle} = \frac{\langle x_{id}, v_{S_i} + m \cdot v_{R} \rangle}{\langle x_{id}, v_R \rangle} = \frac{\langle x_{id}, v_{S_i} \rangle}{\langle x_{id}, v_R \rangle} + m.
\]

Therefore, the decryption result of the probe ciphertext \( C_{S_i} \) is \( m \) if \( id \in S_i \) and \( m' \) if \( id \in S \setminus S_i \). Then we observe that the decryption results of \( C_{S_i} \) and \( C_{S_i-1} \) are the same, as long as we use a secret key for \( id \in S \cup R \) with \( id \neq id_{i-1} \). By the security property of inner product functional encryption, this implies that any coalition of users \( \subseteq S \) cannot distinguish two ciphertexts without having \( sk_{x_{id_{i-1}}} \). 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 that enjoys improved efficiency compared to Agrawal et al. (2016). Let \( \ell \) 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 \( Z \in \mathbb{Z}^{\ell \times n} \), chosen from a Gaussian distribution with standard deviation \( \sigma \). The public key is of the form \( \text{pk} = (A \in \mathbb{Z}_q^{m \times n}, U = ZA \in \mathbb{Z}_q^{\ell \times n}) \). To generate a secret key for the vector \( x \in \mathbb{Z}_p^\ell \), we first pick a vector \( \bar{x} \in \mathbb{Z}^\ell \) from a short Gaussian distribution over \( \mathbb{Z}^\ell \) conditioned on \( \bar{x} \equiv x \mod p \). Then, the secret key is set as \( sk_x = (\bar{x}^t, \bar{x}^t \cdot Z) \). One may wonder why do we set \( \bar{x} \) like this instead of just setting \( x = \bar{x} \). This is because we will use some nice properties of the Gaussian distribution in our security proof, as will be explained later. The ciphertext for a vector \( y \in \mathbb{Z}_p^\ell \) is of the form \( (c_0 \approx As, c_1 \approx Us + p^{k-1} \cdot y) \) where \( x \approx y \) means that \( \|x - y\| \) 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 \( Z \) are small, \( c_1 \approx ZAs \approx Zc_0 \). Given this observation, we can change the distribution of the ciphertext as \( c_0 \) being a random vector \( u \leftrightarrow Z_\ell^\ell \) and \( c_1 \approx Zu + p^{k-1} \cdot y \) without being detected by the adversary, assuming the LWE assumption.

The main difficulty in the proof is in showing that \( c_1 \approx Zu + p^{k-1}y \) does not leak any information more than necessary. Note that \( c_1 \) does leak some information. Namely, given a secret key \( sk_x \) for \( x \), we can still decrypt the modified ciphertext to obtain \( (x, y) \mod p \). What we have to prove is that the ciphertext does not leak any information of \( \langle x', y \rangle \mod p \) for all \( x' \not\in \text{Span}_{\mathbb{Z}_p} \{ \{x_i\}_{i \in [L]} \} \), where \( L \) is the number of key queries and \( \{x_1, \ldots, x_L\} \) 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 \( Z \) is hidden from the adversary. In particular, we explain that an attempt to obtain any information of \( \langle x', y \rangle \mod p \) by computing \( \langle x', c_1 \rangle \approx x'^tZu + p^{k-1} \cdot \langle x', y \rangle \mod q \) fails because \( x'^tZ \) retains sufficiently high min-entropy and thus \( x'^tZu \) is uniformly random modulo \( q \) by the leftover hash lemma.

To see this, let \( X_{top} \in \mathbb{Z}^{L \times \ell} \) be the matrix obtained by vertically concatenating \( \{x_i \in \mathbb{Z}_p^\ell\}_{i \in [L]} \). Via secret keys, the adversary learns the value of \( X_{top}Z \). Let us ignore the additional leakage on \( Z \) from the public key in this overview. Note that in \( X_{top}Z \), the matrix \( X_{top} \) acts in parallel on the columns of \( Z \). We can hence restrict ourselves to the distribution of \( z_i \) conditioned on \( b_i := X_{top}z_i \). It can be seen that \( z_i \) is distributed on the shifted kernel lattice \( \Lambda \), defined as

\[
\Lambda = \{ v \in \mathbb{Z}^m : X_{top} \cdot v = 0 \}.
\]

If the standard deviation \( \sigma \) is sufficiently large (i.e., larger than the smoothing parameter of \( \Lambda \)), the vector \( z_i \) behaves like the continuous Gaussian even though it is sampled from the discrete Gaussian. In particular, it
spreads all directions under the only constraint that $X_{\text{top}} z_i = b_i$, and thus $\langle \mathbf{x}', z_i \rangle$ has sufficiently high entropy, allowing us to conclude. In Agrawal et al. (2016), the equivalent of $X_{\text{top}}$ in their proof is arbitrarily chosen by the adversary and $\ell = L + 1$. This results in exponentially large smoothing parameter for corresponding $\Lambda$. Therefore, they have to take $\sigma$ exponentially large, which is exactly the source of the inefficiency in their scheme. In our case, the matrix $X_{\text{top}}$ 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 $x_i$.) Furthermore, we set $\ell$ large compared to $L$. We can then invoke the result of Agrawal et al. (2013), which says that the smoothing parameter of $\Lambda$ corresponding to such $X_{\text{top}}$ is small. This allows us to choose $\sigma$ 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. We provide a generic transformation from an inner product functional encryption scheme to a traitor tracing scheme that supports multi-message encryption in the full version of this work that is on ePrint Agrawal et al. (2017).

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 $x \leftarrow X$ 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 $(\mathbf{x} \parallel \mathbf{y})$ 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 $\text{Poly}(\cdot)$.

In our scheme descriptions, a user’s identifying information is denoted by $\text{id}$. 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 $R$. The set of traitors is denoted by $T$ and the set of users that are suspected to be traitors is denoted by $S$.

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 $D$. Let $T$ be the set of traitors and we assume that the size $|T|$ of the traitor coalition is at most $t$. The tracing algorithm aims at disclosing the identity of at least one user from the set $T$ of traitors.

In Boneh and Franklin (1999), the minimal black-box access model was considered where the tracing procedure has access to the pirate decryption device $D$ only through an oracle $\mathcal{O}^D$. The oracle $\mathcal{O}^D$ takes as input any message-ciphertext pair $(M, C)$ and returns 1 if $D(C) = M$ 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 $D$ 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 $R$ 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 $C(R)$ be the output of the encryption algorithm $\text{Enc}$ using the public encryption key $\text{pk}$ and $R$ as the set of revoked users. With $C(R)$ as input, the device $D$ outputs $m$ with probability significantly more than $1/|\mathcal{M}|$:

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

for some constant $c > 0$.\(^2\) The probability of decryption for a decoder $D$ can be estimated by repeatedly querying the oracle $\mathcal{O}^D$ 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 Kiayias and Yung (2002)). We also assume that the decoder $D$ 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 Kiayias and Yung (2001b,a).

We let the identity space $\text{ID}$ and the message space $\mathcal{M}$ be implicit arguments to the setup algorithm below. We let the secret key space $K$ 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 $\mathcal{TR} = (\text{Setup}, \text{KeyGen}, \text{Enc}, \text{Dec}, \text{Trace})$ of five probabilistic polynomial-time algorithms with the following specifications.

- **Setup**($1^\lambda, 1^t, 1^r$) takes as input the security parameter $\lambda$, the bound $t$ on the size of traitor coalitions and the bound $r$ on the number of revoked users. It outputs $(\text{msk}, \text{pk}, \text{pd})$ containing the master secret key, the encryption key and the (initially empty) public directory $\text{pd}$. We will implicitly assume that $\text{pd}$ is available to all algorithms.

- **KeyGen**(msk, id) takes as input the master secret msk and an identity id $\in \text{ID}$ of a user, and outputs a secret key $\text{sk}_{\text{id}}$ and some public information $\text{p}_{\text{id}}$ for id. It also updates the public directory $\text{pd}$ to include $\text{p}_{\text{id}}$.\(^3\)

- **Enc**(pk, $R$, m) takes as input the public key pk, a set $R$ of cardinality $\leq r$ which contains the $\text{p}_{\text{id}}$ of each revoked user in pd, and a plaintext message $m \in \mathcal{M}$. It outputs a ciphertext $C \in \mathcal{C}$.

- **Dec**(sk$_{\text{id}}$, C) takes as input a secret key sk$_{\text{id}}$ of a user with identity id and a ciphertext $C \in \mathcal{C}$. It outputs a plaintext $m \in \mathcal{M}$.

- **Trace**(pd, $R$, S, $\mathcal{O}^D$) is a black-box confirmation tracing algorithm that takes as input the public directory pd, a set $R$ of $\leq r$ revoked users, a set $S$ of $\leq t$ suspect users, and has black-box access to the pirate decoder $D$ through the oracle $\mathcal{O}^D$. It outputs an identity id or $\bot$.

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

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

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

\(^2\)In Nishimaki et al. (2016), 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 8 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.

\(^3\)We emphasize that $\text{p}_{\text{id}}$ does not need to contain id.
Public Traceability. It is required that, when $S$ contains the set $T$ of traitors who produced the pirate decoder $D$, then the id output by the tracing algorithm belongs to $T$. This requirement is formalized using the following game, denoted by $\text{AD-\text{TT}}$, between an adversary $\mathcal{A}$ and a challenger:

- The challenger runs $\text{Setup}(1^\lambda, 1^t, 1^r)$ and gives $\mathcal{A}$ $pk$.
- 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 $sk_{id}$. 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 $id \in \mathcal{ID}$ 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 $sk_{id}$. The challenger records the identity query $id$ in a list $T$.
  - Otherwise (i.e., user $id$ is a new user in the system), the challenger runs $\text{KeyGen}(msk, id)$, responds with $sk_{id}$ and updates the directory $pd$ with the public information $p_{id}$ for $id$. The challenger also records the identity query $id$ in the list $T$.
- Adversary $\mathcal{A}$ is allowed to (adaptively) choose a set $R$ of up to $r$ revoked users in $pd$. The challenger gives $\mathcal{A}$ all the corresponding $sk_{id}$. These queries can be interleaved with extensions of the number of users and user corruption queries, in an adaptive manner.
- Adversary $\mathcal{A}$ finally produces a pirate decoder $D$. It chooses a suspect set $S$ of cardinality $\leq t$ that contains $T$, and sends $S$ to the challenger.
- The challenger then runs $\text{Trace}(pd, R, S, OD)$. The adversary wins if both of the following hold:
  - Equation (1) is satisfied for the set of revoked users $R$ chosen by the adversary (i.e., decoder $D$ is useful),
  - the execution of $\text{Trace}$ outputs $\bot$ or outputs an id that does not belong to $T$ with probability $\geq 1/\lambda^c$.

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

Almost Public Traceability. This is the same as public traceability, except that $\text{Trace}$ 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 $\text{Trace}$ to output a $p_{id}$ that does not belong to $pd_T$, which is the set of all $p_{id'}$ for $id' \in T$.

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.

We note that our proposed schemes satisfy the public traceability instead of the almost public traceability. However, it is easy to modify them so that they satisfy the latter. Hereafter, we will not discuss about almost public traceability.

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 IND-CPA security of a trace-and-revoke scheme \( \mathcal{T} \mathcal{R} \) is defined based on the following game.

- The challenger runs \( \text{Setup}(\lambda, 1^t, 1^r) \) and gives the produced public key \( \text{pk} \) 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 \( \mathcal{pd} \). The challenger gives \( \mathcal{A} \) all the \( \text{sk}_id \) such that \( p_d \in \mathcal{R} \).
- 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 \( b \leftarrow \{0, 1\} \) and provides \( C_{mb} \leftarrow \text{Enc}(\text{pk}, \mathcal{R}, m_b) \) 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 \( \text{Adv}^{\text{IND-CPA}}_{\mathcal{T} \mathcal{R}, \mathcal{A}} = |\text{Pr}[b = b'] - 1/2| \). The scheme \( \mathcal{T} \mathcal{R} \) 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 \( \mathcal{FE} = (\mathcal{FE}.\text{Setup}, \mathcal{FE}.\text{KeyGen}, \mathcal{FE}.\text{Enc}, \mathcal{FE}.\text{Dec}) \) of four probabilistic polynomial-time algorithms with the following specifications:

- \( \mathcal{FE}.\text{Setup}(1^\lambda, 1^t) \) takes as input the security parameter \( \lambda \) and outputs the public key and the master secret key pair \((\text{pk}, \text{msk})\);
- \( \mathcal{FE}.\text{KeyGen}(\text{msk}, x) \) takes as input the master secret key \( \text{msk} \) and a vector \( x \in \mathbb{Z}_p^t \) and outputs the secret key \( \text{sk}_x \);
- \( \mathcal{FE}.\text{Enc}(\text{pk}, y) \) takes as input the public key \( \text{pk} \) and a message \( y \in \mathbb{Z}_p^t \) and outputs the ciphertext \( \text{ct}_y \);
- \( \mathcal{FE}.\text{Dec}(\text{sk}_x, \text{ct}_y) \) takes as input the secret key of a user \( \text{sk}_x \) and the ciphertext \( \text{ct}_y \), 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 \((\text{pk}, \text{msk}) \leftarrow \mathcal{FE}.\text{Setup}(1^\lambda, 1^t)\) and \( \forall x, y \in \mathbb{Z}_p^t \):

\[
\mathcal{FE}.\text{Dec}(\mathcal{FE}.\text{KeyGen}(\text{msk}, x), \mathcal{FE}.\text{Enc}(\text{pk}, y)) = \langle x, y \rangle \mod p.
\]

**Security of \( \mathcal{FE} \).** We consider security of functional encryption in the standard indistinguishability setting Boneh et al. (2011).

**Definition 3** A functional encryption scheme \( \mathcal{FE} = (\mathcal{FE}.\text{Setup}, \mathcal{FE}.\text{KeyGen}, \mathcal{FE}.\text{Enc}, \mathcal{FE}.\text{Dec}) \) 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 \( \mathcal{FE}.\text{Setup}(1^\lambda, 1^t) \) and the master public key \( \text{mpk} \) is given to \( \mathcal{A} \).
- The adversary adaptively makes secret key queries to the challenger. At each query, adversary \( \mathcal{A} \) chooses a vector \( x \in \mathbb{Z}_p^t \) and obtains the corresponding secret key \( \text{sk}_x \leftarrow \mathcal{FE}.\text{KeyGen}(\text{msk}, x) \).
exists a probabilistic polynomial-time algorithm which is within statistical distance 2 of parameter Λ,

denoted by A, against such a restricted adversary will be called 1-IND-CPA.

2.3 Lattice background

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

$$\eta_\varepsilon(\Lambda) = \min \left( \sigma > 0 : \sum_{b \in \hat{\Lambda}} \exp(-\pi\|\hat{\Lambda}^T \cdot b\|^2/\sigma^2) \leq 1 + \varepsilon \right),$$

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

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

For m ≥ n and a rank-n matrix X ∈ R^{m×n}, denote \( U_X = \{ \| Xu \| : u \in \mathbb{R}^n, \| u \| = 1 \} \). The least singular value of X is then defined as \( s_n(X) := \inf(U_X) \) and similarly the largest singular value of X is \( s_1(X) := \sup(U_X) \). For a matrix Y ∈ R^{n'×m'} with n' > m', the least singular value and the largest singular value are defined as \( s_1(Y) := s_1(Y^T) \) and \( s_m(Y) := s_{m'}(Y^T) \) 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 1 (Corollary 2.8 in Gentry et al. (2008)) Let \( \Lambda' \subseteq \Lambda \subseteq \mathbb{R}^m \) be two lattices with the same dimension. Let \( \varepsilon \in (0, 1/2) \). Then for any \( c \in \mathbb{R}^m \) and any \( \Sigma \) such that \( s_m(\Sigma) \geq \eta_\varepsilon(\Lambda') \), the distribution \( D_{\Lambda,\Sigma,c} \mod \Lambda' \) is within statistical distance 2ε from the uniform distribution over \( \Lambda/\Lambda' \).

Lemma 2 (Lemma 1 in Katzumata and Yamada (2016)) Let \( r \geq \Omega(\sqrt{\Lambda}) \) and \( q, \ell, m > 0 \) integers. Let \( b \in \mathbb{Z}_q^m \) be arbitrary and \( x \) chosen from \( D_{\mathbb{Z}^m, r} \). Then for any \( V \in \mathbb{Z}_q^{\ell \times m} \) and positive real \( r' > s_1(V) \), there exists a probabilistic polynomial-time algorithm \( \text{ReRand}(V, b + x, r, r') \) that outputs \( b' = Vb + x' \in \mathbb{Z}_q^\ell \) where \( x' \) is within statistical distance \( 2^{-\Omega(\lambda)} \) from \( D_{\mathbb{Z}_q^{\ell \times 2r'}} \).

We use the following variant of the leftover hash lemma, adapted from Micciancio and Mol (2011) (see also Lemma 11 in Agrawal et al. (2016)).
Lemma 3 (Micciancio and Mol (2011)) Let \( m \geq n \geq 1 \) and \( q = p^k \) for \( p \) prime and \( k \geq 1 \). Take \( X \) 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 \( (A, A \cdot x) \), where sampling \( A \leftarrow \mathbb{Z}_q^{n \times m} \) and \( x \leftarrow X \). Then,

\[
\Delta(D_0, D_1) \leq \frac{1}{2} \sqrt{\sum_{i=1}^{k} p^{i-n} \cdot Pr_i},
\]

where \( Pr_i \) is the collision probability of two independent samples from \( (X \mod p^i) \).

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

\[
\Delta(D_0, D_1) \leq \varepsilon + \sqrt{q^n/p^m}.
\]

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

Lemma 4 (Special case of Lemma 8 in Agrawal et al. (2016)) There exists a universal constant \( K > 1 \) such that for all \( m \geq 2n \), \( \varepsilon > 0 \) and \( \sigma \geq K \eta_s(\mathbb{Z}) \), the following holds for \( X \leftarrow D_{\lambda, \sigma}^{n \times m} : \)

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

We will also require the following theorem, adapted from Theorem 17 in Ling et al. (2014).

Theorem 1 (Ling et al. (2014)) Let \( n, m_1, m_2 \), and \( \lambda \) be integers satisfying \( m_2 \geq m_1 > 100 \) and \( \sigma_1, \sigma_2 \) be positive real numbers. Let \( n' = \max(\lambda, n) \) and assume that \( n' > 100 \). We also assume that they satisfy

\[
\sigma_1 \geq \Omega(\sqrt{m_1 n' \log m_1}), m_1 \geq \Omega(n' \log(\sigma_1 n')) \text{ and } \sigma_2 \geq \Omega(n'^{5/2} \sqrt{m_1 \sigma_1^3 \log^{3/2}(m_1 \sigma_1)}).
\]

Then, there exists a probabilistic polynomial-time algorithm that given \( n, m_1, m_2, \lambda \) (in unary), \( \sigma_1 \), and \( \sigma_2 \), returns \( X_1 \in \mathbb{Z}_n^{m_1}, X_2 \in \mathbb{Z}_n^{m_2} \), and \( U \in \mathbb{Z}_n^{m_1} \) with \( m = m_1 + m_2 \) such that:

- the distribution of \((X_1, X_2)\) is within statistical distance \( 2^{-\Omega(n')} \) of the distribution \( D_{\lambda, \sigma_1}^{n \times m_1} \times (D_{\mathbb{Z}_n^{m_2}, \sigma_2, \delta_i}^{m_2})^l \), where \( \delta_i \) denotes the \( i \)th canonical unit vector in \( \mathbb{Z}_n^{m_2} \) whose \( i \)th coordinate is \( 1 \) and whose remaining coordinates are \( 0 \),
- we have \( |\det U| = 1 \) and \((X_1|X_2) \cdot U = (I_n|0)\),
- every column of \( U \) 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 Ling et al. (2014). This is just for a notational convenience. Secondly, the distribution of \( X = (X_1|X_2) \) in Theorem 17 in Ling et al. (2014) is slightly different from the above in that all entries of the first column of \( X \) equal to \( 1 \). As noted right after Lemma 7 in Ling et al. (2014), the theorem still holds even with the change. Finally, in the above theorem, we introduce the statistical security parameter \( \lambda \) and differentiate it from the lattice dimension \( n \), while the theorem in Ling et al. (2014) 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 \( X \) is chosen from a slightly different distribution and \( U \) need not be efficiently sampleable.

Lemma 5 Let \( n, m_1, m_2, m, \lambda, n', \sigma_1, \sigma_2 \) be as in Theorem 1. Then, for all but \( 2^{-\Omega(n')} \) probability over \((X_1, X_2) \in \mathbb{Z}_n^{m_1} \times \mathbb{Z}_n^{m_2} \) chosen from \( D_{\lambda, \sigma_1}^{n \times m_1} \times D_{\mathbb{Z}_n^{m_2}, \sigma_2}^{n \times m_2} \), there exists \( U \in \mathbb{Z}_n^{m_1} \) such that \( |\det U| = 1 \), \((X_1|X_2) \cdot U = (I_n|0)\), and every column of \( U \) has norm \( \leq O(\sqrt{n' m_1 \sigma_2}) \).
To prepare for the proof of Lemma 5, we define Rényi Divergence (RD) and review its properties following Bai et al. (2015). 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 6 (Lemma 2.9 in Bai et al. (2015))** Let \( P \) and \( Q \) denote distributions with \( \text{Supp}(P) \subseteq \text{Supp}(Q) \) and \( A \subseteq \text{Supp}(Q) \) 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 7 (Lemma 4.2 in Langlois et al. (2014))** For any \( n \)-dimensional lattice \( L \subseteq \mathbb{R}^n \) and invertible matrix \( \Sigma \), set \( P = D_{\Lambda, \Sigma, w} \) and \( Q = D_{\Lambda, \Sigma, z} \) for some fixed \( w, z \in \Lambda \). Then, \( R(P\|Q) \leq \exp(2\pi \|w - z\|^2/s_\Sigma(\Sigma)^2) \).

Then, we proceed to the proof of Lemma 5.

**Proof:** [Proof of Lemma 5] Let \( A \subseteq \mathbb{Z}^{n \times m} \) be the set of \( X = (X_1|X_2) \) such that \( U \) satisfying the properties listed in the statement does not exist. Theorem 1 implies that when \( X \) is sampled from the distribution \( Q := \sum_{\delta_1} D_{\delta}(\delta_1) \times \sum_{\delta_2} D_{\delta}(\delta_2) \), we have \( Q(A) \leq 2^{-\Omega(n)} \). We want to prove that \( P(A) = 2^{-\Omega(n)} \) for the distribution \( P := D_{\delta}(\delta_1) \times D_{\delta}(\delta_2) \). By Lemma 6, 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

\[
R(P||Q) = R(D_{\delta, \sigma_1}^{n \times m_1} \times D_{\delta, \sigma_2}^{n \times m_2} || D_{\delta, \sigma_1}^{n \times m_1} \times D_{\delta, \sigma_2}^{n \times m_2} \\
\quad \quad \quad \quad \times (D_{\delta, \sigma_1}^{n \times m_1} \times \cdots \times D_{\delta, \sigma_n}^{n \times m_n})) \\
\quad \quad \quad \leq \exp(2\pi n/\sigma_2^2),
\]

where we use Lemma 7 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 5.

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 Regev (2005); Peikert (2009); Brakerski et al. (2013).

**Definition 4** 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 LWE\(_{n,m,q,\alpha} \) of \( \mathcal{A} \) is defined as follows:

\[
| \Pr[\mathcal{A}(A, As + x) \to 1] - \Pr[\mathcal{A}(A, w + x) \to 1] |
\]

where \( A \leftarrow \mathbb{Z}_q^{n \times m}, s \leftarrow \mathbb{Z}_q^n, w \leftarrow \mathbb{Z}_q^m, \) and \( x \leftarrow D_{\delta, q}^m \). We say that 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.
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. Setup\((1^\lambda, 1^t, 1^r)\). Upon input the security parameter \(\lambda\), the bound \(t\) on the number of traitors and the bound \(r\) on the number of revoked users, proceed as follows:
   (a) Let \((pk, msk) \leftarrow \mathcal{FE}.\text{Setup}(1^\lambda, 1^t)\), where \(\ell = t + r + 1\).
   (b) Output the public key \(pk\) and master secret key \(msk\).

2. KeyGen\((msk, id)\). Upon input the master secret key \(msk\) and a user identity \(id \in ID\), proceed as follows:
   (a) Sample \(x_{id} \leftarrow \mathbb{Z}_p^\ell\). The pair \((id, x_{id})\) is appended to the public directory \(pd\).
   (b) Let \(sk_{id} \leftarrow \mathcal{FE}.\text{KeyGen}(msk, x_{id})\).
   (c) Output \(sk_{id}\).

3. Enc\((pd, pk, R, m)\). Upon input the public key \(pk\), a set of revoked users \(R\) of cardinality \(\leq r\) and a plaintext messages \(m \in M = \mathbb{Z}_p\), proceed as follows:
   (a) Compute \(v_R \in \mathbb{Z}_p^\ell \setminus \{\vec{0}\}\) such that \((x_{id}, v_R) = 0\) for every \(id \in R\).
   (b) Compute \(y_R = m \cdot v_R\).
   (c) Output \(C = (C_1, C_2) = (\mathcal{FE}.\text{Enc}(pk, y_R), R)\).

4. Dec\((pd, sk_{id}, C)\). Upon input the secret key \(sk_{id}\) for user \(id\) and a ciphertext \(C = (C_1, C_2)\), proceed as follows:
   (a) Parse \(C_2\) as \(C_2 = R\). If \(id \in R\), then abort.
   (b) Compute \(v_R \in \mathbb{Z}_p^\ell \setminus \{\vec{0}\}\) such that \((x_{id}, v_R) = 0\) for every \(id \in R\).
   (c) Compute and output \(m = \mathcal{FE}.\text{Dec}(sk_{id}, C_1)/(x_{id}, v_R)\).

5. Trace\((pd, R, S, O^D)\). Upon input the public directory \(pd\), a revoked set of users \(R\), a suspect set \(S\) of users and given access to the oracle \(O^D\), first proceed as follows:
   (a) Find \(m, m' \in M\) such that the following quantity is non-negligible:
   \[
   \left| \begin{array}{c}
   \Pr_{C \leftarrow \text{Enc}(pd, pk, R, m)} [O^D(C, m) = 1] \\
   - \Pr_{C' \leftarrow \text{Enc}(pd, pk, R, m')} [O^D(C', m) = 1]
   \end{array} \right|.
   \]
   (b) Set \(S_i = \{id_1, id_2, \ldots\} = S \setminus R\).
   (c) Compute \(v_R \in \mathbb{Z}_p^\ell \setminus \{\vec{0}\}\) such that \((x_{id}, v_R) = 0\) for every \(id \in R\).

Then execute the following steps with \(i = 1, 2, \ldots\):
   (d) If \(i = 1\), set \(v_{S_i} = \vec{0}\). If \(S_i = \emptyset\), set \(v_{S_i} = (m' - m) \cdot v_R\). Else compute \(v_{S_i} \in \mathbb{Z}_p^\ell\) such that:
      i. \((x_{id}, v_{S_i}) = 0\) for every \(id \in S_i \cup R\).
Assume that longer, but make the encryption and decryption algorithms slightly more efficient.

Let \( C \in \mathbb{Z}_p \) and \( y \in \mathbb{Z}_p \). We proceed in the same way (using always the same unique way across all algorithms. One way of achieving this property is to order the 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 \( m \).

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

We remark that one can send \( R \) instead of \( v_R \) in the encryption algorithm. This will make the ciphertext longer, but make the encryption and decryption algorithms slightly more efficient.

We first check the correctness of the scheme.

**Theorem 2** Assume that \( p = \lambda^{\omega(1)} \). Let \( R \) be a set of revoked users of cardinality \( \leq r \). Then, for every \( id \notin R \) and every \( m \in M = \mathbb{Z}_p \), we have

\[
\text{Dec}(pd, sk_{id}, \text{Enc}(pd, pk, R, m)) = m,
\]

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

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

\[
\text{Dec}(pd, sk_{id}, C) = \frac{\mathcal{FE}.\text{Dec}(sk_{id}, C_1)}{\langle x_{id}, v_R \rangle} = \frac{\langle x_{id}, m \cdot v_R \rangle}{\langle x_{id}, v_R \rangle} = m,
\]

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 \( O^D \). 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 \( S_i = \emptyset \) and \( v_{S_i} = (m' - m) \cdot v_R \). Consequently, we have \( C_{S_i} = (\mathcal{FE}.\text{Enc}(pk, y), R) \) where \( y = v_{S_i} + m \cdot v_R = m' \cdot v_R \).) The fact that \( O^D \) 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 \( O^D \) 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 \( O^D \) only achieves correct decryption with non-negligible advantage.

**Lemma 8** Let \( R \) be arbitrary and assume that Equation (1) holds for \( R \). Then, with probability \( \geq 1/(4\lambda^c) \) over the choice of \( m, m' \leftrightarrow M \), we have:

\[
\Pr_{C \leftarrow \text{Enc}(pk, R, m)} [O^D(C, m) = 1] - \Pr_{C' \leftarrow \text{Enc}(pk, R, m') [O^D(C', m) = 1]} \geq \frac{1}{2\lambda^c}.
\]
Based on Lemma 8, Step (a) of Algorithm Trace can be implemented by repeatedly sampling \( m, m' \leftarrow \mathcal{M} \) and estimating the probabilities that \( O^D(C, m) = 1 \) and \( O^D(C', m) = 1 \) 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 \( O^D(C', m) = 1 \), where \( C' \leftarrow \text{Enc}(pd, pk, R, m') \). Equation (1) states that

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

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

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

(2)

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

\[
\Pr_{m' \leftarrow \mathcal{M}} \Pr_{m \leftarrow \mathcal{M}} [P(m, m) - P(m', m)] < \frac{1}{2\lambda c} > 1 - \frac{1}{2\lambda c} > \frac{1}{2}.
\]

(3)

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

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

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

\[
\Pr_{m, m'} [P(m, m) - P(m', m)] < \frac{1}{2\lambda c} = \Pr_{m'} [\Pr_{m} [P(m, m) - P(m', m)] < \frac{1}{2\lambda c}] = \frac{1}{|\mathcal{M}|} \sum_{m'} \Pr_{m} [P(m, m) - P(m', m)] < \frac{1}{2\lambda c}]
\]

\[
+ \frac{1}{|\mathcal{M}|} \sum_{m'} \Pr_{m} [P(m, m) - P(m', m)] < \frac{1}{2\lambda c}
\]

\[
\leq \frac{1}{|\mathcal{M}|} \sum_{m'} 1 + \frac{1}{|\mathcal{M}|} \sum_{m'} (1 - \frac{1}{2\lambda c})
\]

\[
= x + (1 - x)(1 - \frac{1}{2\lambda c}) \leq 1 - \frac{1}{4\lambda c},
\]

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 3 If $\mathcal{F}\mathcal{E}$ is $r$-IND-CPA secure, then $\mathcal{T}\mathcal{R}$ is IND-CPA secure.

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

- It first obtains the public key $\textsf{pk}$ output by the $\mathcal{F}\mathcal{E}$ challenger (who runs the $\mathcal{F}\mathcal{E}.\textsf{Setup}(1^\lambda, 1^t)$ algorithm) and relays it to $\mathcal{A}_{\mathcal{T}\mathcal{R}}$.

- The adversary $\mathcal{A}_{\mathcal{T}\mathcal{R}}$ adaptively chooses at most $r$ identities $\text{id}$ (that forms the revoked set $\mathcal{R}$) and are included in $\text{pd}$. The adversary $\mathcal{A}_{\mathcal{F}\mathcal{E}}$ then queries the $\mathcal{F}\mathcal{E}$ challenger for each $\text{x}_{\text{id}}$ for all $\text{id} \in \mathcal{R}$ and receives the corresponding $\text{sk}_{\text{id}}$. Adversary $\mathcal{A}_{\mathcal{F}\mathcal{E}}$ relays all $\text{sk}_{\text{id}}$ for each $\text{id} \in \mathcal{R}$ to $\mathcal{A}_{\mathcal{T}\mathcal{R}}$.

- When $\mathcal{A}_{\mathcal{T}\mathcal{R}}$ chooses two messages $m_0, m_1 \in \mathcal{M}$ and provides them to $\mathcal{A}_{\mathcal{F}\mathcal{E}}$, adversary $\mathcal{A}_{\mathcal{F}\mathcal{E}}$ proceeds as follows:
  - It computes $\textbf{v}_{\mathcal{R}} \in \mathbb{Z}_p^t \setminus \{\vec{0}\}$ such that $\langle \text{x}_{\text{id}}, \textbf{v}_{\mathcal{R}} \rangle = 0$ for every $\text{id} \in \mathcal{R}$.
  - It sends $y_{\mathcal{R},0} = m_0 \cdot \textbf{v}_{\mathcal{R}}$ and $y_{\mathcal{R},1} = m_1 \cdot \textbf{v}_{\mathcal{R}}$ to the $\mathcal{F}\mathcal{E}$ challenger who samples $b \leftarrow \{0, 1\}$ and encrypts $y_{\mathcal{R},b}$ as $C_{y_{\mathcal{R},b}} \leftarrow \mathcal{F}\mathcal{E}.\text{Enc}(\text{pk}, y_{\mathcal{R},b})$.
  - Adversary $\mathcal{A}_{\mathcal{F}\mathcal{E}}$ receives $C_{y_{\mathcal{R},b}}$ from the $\mathcal{F}\mathcal{E}$ challenger and sends $C = (C_{y_{\mathcal{R},b}}, \mathcal{R})$ to $\mathcal{A}_{\mathcal{T}\mathcal{R}}$.

- Finally, adversary $\mathcal{A}_{\mathcal{T}\mathcal{R}}$ outputs its guess $b' \in \{0, 1\}$ and $\mathcal{A}_{\mathcal{F}\mathcal{E}}$ also outputs $b'$ as its own guess of $b$.

Note that adversary $\mathcal{A}_{\mathcal{F}\mathcal{E}}$ behaves as an IND-CPA challenger in the view of $\mathcal{A}_{\mathcal{T}\mathcal{R}}$. Further, it is a valid adversary against $\mathcal{F}\mathcal{E}$, as $(y_{\mathcal{R},0}, x_{\text{id}}) = (y_{\mathcal{R},1}, x_{\text{id}})$ for every vector $x_{\text{id}}$ queried to the $\mathcal{F}\mathcal{E}$ challenger (i.e., each $\text{id} \in \mathcal{R}$). The advantage of $\mathcal{A}_{\mathcal{F}\mathcal{E}}$ is exactly the same as the advantage of $\mathcal{A}_{\mathcal{T}\mathcal{R}}$. \hfill $\square$

We may observe that for $\mathcal{T}\mathcal{R}$ to be IND-CPA secure, an $r$-IND-CPA secure $\mathcal{F}\mathcal{E}$ 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 $\mathcal{F}\mathcal{E}$ scheme that is $(t + r)$-IND-CPA secure.

3.3 Traceability

Here, we prove the traceability of the scheme. To start with, we first prove the following lemma.

Lemma 9 Assume that a pirate decoder $\mathcal{D}$ satisfies Equation (1) for some $\mathcal{R}$ and $\mathcal{S}$. Then, the execution of Trace does not return $\bot$ but returns some $\text{id} \in \mathcal{S}$ 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 $\text{id}$. We consider the probabilities $p_i$ at the start and end of that modified execution.

1. At the beginning, algorithm Trace considers $S_1 = S \setminus \mathcal{R}$ and $v_{S_1} = \vec{0}$. Hence, the genuine ciphertext output by the $\text{Enc}$ algorithm and the probe ciphertext created by the Trace algorithm for the suspect subset $S_1$ are exactly the same.

2. When $i = |S \setminus \mathcal{R}| + 1$, we have $S_i = \emptyset$ and $v_{S_i} = (m' - m) \cdot v_{\mathcal{R}}$. In Step (a) of the Trace algorithm, the messages $m$ and $m'$ were chosen such that the difference in the probabilities $p_1$ and $p_{|S \setminus \mathcal{R}|+1}$ is $\geq 1/(2\lambda')$. 

16
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, \textbf{Trace} algorithm outputs \( id_{i-1} \) with overwhelming probability. \( \square \)

Then, we prove the following theorem.

\textbf{Theorem 4} \textit{If }\( \mathcal{FE} \text{ is } (t + r)\text{-IND-CPA secure, then } \mathcal{T} \mathcal{R} \text{ satisfies public traceability.} \)

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

1. It first obtains the public key \( \text{pk} \) output by the \( \mathcal{FE} \) challenger (who runs the \( \mathcal{FE}.\text{Setup}(1^\lambda, 1^t) \) algorithm) and relays it to the adversary \( A \).

2. When \( A \) asks \( A_{\mathcal{FE}} \) to create a \( p_{id} \) for some \( id \), adversary \( A_{\mathcal{FE}} \) in turn asks the \( \mathcal{FE} \) challenger to do the same. The \( \mathcal{FE} \) challenger randomly chooses a vector \( x_{id} \leftarrow \mathbb{Z}_p^t \) and sends it to \( A_{\mathcal{FE}} \) who further relays it to \( A \).

3. When \( A \) makes a key query for an identity \( id \), adversary \( A_{\mathcal{FE}} \) queries the \( \mathcal{FE} \) challenger for a secret key. Adversary \( A_{\mathcal{FE}} \) receives the corresponding \( sk_{id} \) from the \( \mathcal{FE} \) challenger and relays it to \( A \).

4. When \( A \) chooses a set \( \mathcal{R} \) of up to \( r \) revoked users, adversary \( A_{\mathcal{FE}} \) makes \( |\mathcal{R}| \) key queries to the \( \mathcal{FE} \) challenger. Adversary \( A_{\mathcal{FE}} \) is given the set \( sk_{id} \)'s of corresponding secret keys that is relayed to \( A \). Recall that by the definition of the public traceability game, these queries can be interleaved with extensions of the number of users and user corruption queries, in an adaptive manner.

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

5. Adversary \( A \) finally produces a pirate decoder \( \mathcal{D}^4 \) and chooses a suspect set \( S \) of cardinality \( \leq t \) that contains \( T \). Then, the adversary \( A_{\mathcal{FE}} \) executes the \textbf{Trace} algorithm on \( \mathcal{O}^D \) to find \( i \) such that \( |p_i - p_{i-1}| \) is non-negligible. If \textbf{Trace} outputs \( \bot \) or index \( i \) such that \( id_{i-1} \in \mathcal{T} \), then \( A_{\mathcal{FE}} \) outputs a random bit. We say that the event \textbf{Abort} occurs in such a case. Otherwise, it sets \( y_0 = v_{S_{i-1}} + m \cdot v \mathcal{R} \) and \( y_1 = v_{S_i} + m \cdot v \mathcal{R} \), and sends them as challenge messages to the \( \mathcal{FE} \) challenger.\( ^5 \)

6. The \( \mathcal{FE} \) challenger samples \( b \leftarrow \{0, 1\} \) and then sends \( \mathcal{FE}.\text{Enc}(\text{pk}, y_b) \) to \( A_{\mathcal{FE}} \). The adversary \( A_{\mathcal{FE}} \) runs \( \mathcal{O}^D \) on input \( (C_b, m) \), where \( C_b = (\mathcal{FE}.\text{Enc}(\text{pk}, y_b), \mathcal{R}) \). Then \( \mathcal{O}^D \) outputs the bit \( b' \in \{0, 1\} \).

7. Finally, adversary \( A_{\mathcal{FE}} \) outputs the same bit \( b' \in \{0, 1\} \) if \( p_i - p_{i-1} > 0 \) and \( 1 - b' \) otherwise.

We first argue that \( A_{\mathcal{FE}} \) is a valid adversary against the \( \mathcal{FE} \) challenger. Recall that when \textbf{Abort} does not occur, we have \( id_{i-1} \notin \mathcal{T} \) but \( id_{i-1} \in \mathcal{S} \). The keys queried by \( A_{\mathcal{FE}} \) are for \( id \in \mathcal{R} \cup \mathcal{T} \). This set \( \mathcal{R} \cup \mathcal{T} \) can be partitioned into \( \mathcal{R} \cup (\mathcal{T} \cap S_{i-1}) \) and \( \mathcal{T} \cap (S_i \setminus S_{i-1}) \). Note that since \( id_{i-1} \notin \mathcal{T} \), we have \( \mathcal{T} \cap S_{i-1} = \mathcal{T} \cap S_i \) and thus \( \mathcal{R} \cup (T \cap S_{i-1}) = \mathcal{R} \cup (T \cap S_i) \).

1. For \( id \in \mathcal{R} \), we have \( \langle x_{id}, y_0 \rangle = \langle x_{id}, y_1 \rangle = 0 \). For \( id \in S_{i-1} \cap \mathcal{T} = S_i \cap \mathcal{T} \), we have \( \langle x_{id}, y_0 \rangle = \langle x_{id}, y_1 \rangle = m \cdot \langle x_{id}, v \mathcal{R} \rangle \). Hence for all \( id \in \mathcal{R} \cup (T \cap S_{i-1}) \) for which the \( sk_{id} \) was queried by \( A_{\mathcal{FE}} \), the inner products \( \langle x_{id}, y_0 \rangle \) and \( \langle x_{id}, y_1 \rangle \) have the same value.

\( ^4 \)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), \( v \mathcal{R} \in \mathbb{Z}_p^t \) is chosen as in Step (c), and \( v_{S_{i-1}}, v_{S_i} \in \mathbb{Z}_p^t \) are chosen as in Step (d) of algorithm \textbf{Trace}.
2. Similarly, for \( id \in T \cap (S_1 \setminus S_{i-1}) \), we have \( \langle x_{id}, y_0 \rangle = \langle x_{id}, y_1 \rangle = m' \cdot \langle x_{id}, v_R \rangle \).

Hence, \( A_{FE} \) is a valid adversary against the \( FE \) challenger.

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

- In the above game, \( Abort \) does not occur with non-negligible probability.
- Conditioned on \( Abort \) not occurring, \( Trace \) outputs \( id_{i-1} \) such that \( |p_i - p_{i-1}| \) is non-negligible.

Indeed, the combination of them implies that the advantage of \( A_{FE} \) is non-negligible, since \( |p_i - p_{i-1}| \) is the advantage of \( A_{FE} \) conditioned on \( Abort \) not occurring.

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 \( D \) satisfies the winning condition, when \( Trace \) is executed on input \( O^D \), it outputs \( \perp \) or it outputs an \( id_{i-1} \notin T \) with probability at least \( 1/\lambda^c \). The claim now follows since the former event occurs only with negligible probability by Lemma 9. \( \square \)

4 Trace and Revoke from Learning with Errors

Recall that Agrawal et al. (2016) 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 Agrawal et al. (2016). 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 Agrawal et al. (2016), 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 Agrawal et al. (2016), 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 \( p = 2^{O(\lambda)} \), 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 \( \mathbb{Z}_p^t \), and apply the transformation from the previous section.

The resulting trace-and-revoke scheme inherits the unsatisfactory performance of its underlying IPFE (see Agrawal et al., 2016, 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 us to take smaller parameters.\(^6\)

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

### 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. Our construction is quite close to Agrawal et al. (2016) except the key generation algorithm. In Agrawal et al. (2016), the key generation algorithm is deterministic, whereas in our scheme it is randomized and involves certain Gaussian distribution. This change allows us to prove the security of our scheme in the improved parameter setting compared with Agrawal et al. (2016).

**Construction.** Let \( p \) be the modulus of the scheme, \( 2\ell \) be the dimension of the scheme, and \( L \) be the upper bound on the size of the collusion.

- \( \mathcal{FE}.\text{Setup}(1^\lambda, 1^L, 1^{2\ell}) \). 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 \( A \leftarrow \mathbb{Z}_q^{m \times n} \) and \( Z \leftarrow D_{\mathbb{Z}, \sigma_0}^{2\ell \times m} \). Compute \( U = Z \cdot A \in \mathbb{Z}_q^{2\ell \times n} \). Define
  \[
  \text{msk} := Z \quad \text{and} \quad \text{pk} := (A, U).
  \]

- \( \mathcal{FE}.\text{KeyGen}(\text{msk}, x) \). Given \( x = (x_1, \ldots, x_{2\ell})^\ell \in \mathbb{Z}_p^{2\ell} \), sample \( \bar{x}_i \leftarrow D_{p\mathbb{Z} + x_i, \sigma_1} \) for \( i \in [\ell] \) and \( \bar{x}_i \leftarrow D_{p\mathbb{Z} + x_i, \sigma_2} \) for \( i \in [\ell + 1, 2\ell] \). Set \( \bar{x} := (\bar{x}_1, \ldots, \bar{x}_{2\ell}) \in \mathbb{Z}_p^{2\ell} \) and \( z_x^\ell = \bar{x}^t \cdot Z = \mathbb{Z}^m \). Note that we have \( \bar{x} \equiv x \mod p \) by construction. Finally, return \( \bar{x}_x = (\bar{x}, z_x) \).

- \( \mathcal{FE}.\text{Enc}(pk, y) \). To encrypt a vector \( y \in \mathbb{Z}_p^{2\ell} \), sample \( s \leftarrow \mathbb{Z}_q^n \), \( e_0, e_1 \leftarrow D_{\mathbb{Z}, \alpha q}^{m} \) and compute
  \[
  c_0 = A s + e_0 \in \mathbb{Z}_q^m, \quad c_1 = U s + e_1 + p^{k-1} \cdot y \in \mathbb{Z}_q^{2\ell}.
  \]
  Then, return the ciphertext \( C = (c_0, c_1) \).

- \( \mathcal{FE}.\text{Dec}(\text{sk}_id, C) \). Given \( C = (c_0, c_1) \) and a secret key \( (\bar{x}, z_x) \) for \( x \in \mathbb{Z}_p^{2\ell} \), compute \( \mu' = (\bar{x}, c_1) - (z_x, c_0) \mod q \) and output the value \( \mu \in \mathbb{Z}_p^t \) that minimizes \( |p^{k-1} \cdot \mu - \mu'| \).

\(^6\)We observe that the scheme from Subsection 4.2 allows for polynomial error rates, but the correctness of our trace-and-revoke construction requires \( p \geq \lambda^{1.0} \), which leads to a \( \lambda^{1.0} \) LWE error rate in our IPFE.
Setting the Parameters. We have to set the parameters so that the correctness requirement is satisfied and the security reduction from $\text{LWE}_{n,m,q,\alpha'}$ works, for some non-trivial error rate $\alpha'$. 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 1 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 2 in the security proof,
- $\kappa \geq \Omega(\lambda + L \log \lambda)$, to ensure the (overwhelmingly likely) existence of a $U$ as in Lemma 5 in the security proof,
- $\sigma_1 \geq \Omega(\sqrt{\ell \kappa \log \ell})$, $\ell \geq \Omega(\kappa \log(\sigma_1 \kappa))$, and $\sigma_2 \geq \Omega(\kappa^{3/2} \ell \sigma_1^2 \log^{3/2}(\ell \sigma_1))$, to be able to apply Lemma 5 in the security proof with $\kappa \geq \Omega(\lambda + L \log \lambda)$ as above,
- $\sigma_0 \geq \Omega(pk \ell \sigma_2)$ and $q^{n+1}/p^m \leq 2^{-\Omega(\kappa)}$, to be able to apply Lemma 10 in the security proof.

To satisfy the above requirements and rely on LWE parameters for which all known attacks cost $2^{\omega(\lambda)}$, we may set the parameters as follows. We choose $\kappa = \Theta(\lambda + L \log \lambda)$, $p = \lambda^{\omega(1)}$, and:

$$
\ell = \tilde{\Theta}(\lambda + L \log p), \quad 
\sigma_0 = \tilde{\Theta}(\lambda + L)^6(p \log p)^3 \lambda, \quad 
\sigma_1 = \Theta(p \sqrt{\lambda}), \quad 
\sigma_2 = \tilde{\Theta}(\lambda + L)^3(p \log p)^2 \lambda, \quad 
1/\alpha = \tilde{\Theta}(\lambda + L)^9(p \log p)^6 \lambda^2, \quad 
1/\alpha' = \tilde{\Theta}(\lambda + L)^{14.5}(p \log p)^9 \lambda^3, \quad 
m = \tilde{\Theta}(\lambda + L), \quad 
q = \tilde{\Theta}(\lambda + L)^{15}(p \log p)^9 \lambda^3, \quad 
k = \Theta(1), \quad 
n = \Theta(\lambda).
$$

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

$$
\mu' = \langle \bar{x}, e_1 \rangle - \langle z_x, e_0 \rangle = p^{k-1} \cdot \langle x, y \rangle + \langle \bar{x}, e_1 \rangle - \langle z_x, e_0 \rangle.
$$

Below, we show that the magnitude of the term $\langle \bar{x}, e_1 \rangle - \langle z_x, e_0 \rangle$ 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 $D_{p\bar{z}+x_1,\sigma_1}$ if $i \in [\ell]$ and $D_{p\bar{z}+x_1,\sigma_2}$ otherwise. We thus have $\|\bar{x}\| \leq (\sigma_1 + \sigma_2)\sqrt{\ell} \cdot \omega(\sqrt{\log \lambda})$ with probability $1 - \lambda^{-\omega(1)}$. This, together with $e_1 \sim D_{\lambda, q'}$, implies that $|\langle \bar{x}, e_1 \rangle| \leq \sigma_0(\sigma_1 + \sigma_2)\sqrt{\ell} \cdot \omega(\log \lambda)$ with probability $1 - \lambda^{-\omega(1)}$. Furthermore, since each column of $Z$ is chosen from $D_{\lambda, \sigma_0}$, we have $\|z_x\| \leq \sigma_0(\sigma_1 + \sigma_2)\sqrt{\ell m} \cdot \omega(\log \lambda)$ with probability $1 - \lambda^{-\omega(1)}$. As a result, we have $|\langle z_x, e_0 \rangle| \leq \sigma_0(\sigma_1 + \sigma_2)\alpha q \sqrt{\ell m} \cdot \omega(\log^{3/2} \lambda)$ 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. The proof is similar to Agrawal et al. (2016), but we exploits the weaker security model of bounded number of random key queries. In particular, we perform a much more careful analysis on the conditional distribution of $Z$ from the view of the adversary.

**Theorem 5** If the parameters are set as above, the above scheme is L-IND-CPA secure under the LWE$\text{LWE}_{n,m,q,\alpha'}$ assumption.

---

7We 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.
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 $A$ is given the master public key $\mathcal{pk}$. Then, it sees $Q$ random vectors $\{x_i\}_{i \in \{0,1\}}$, where $x_i \leftarrow \mathbb{Z}_p^2$. Then, the adversary is allowed to make key queries for random vectors chosen as $x \leftarrow \mathbb{Z}_p^2$. In the challenge phase, the adversary $A$ comes up with two distinct vectors $y_0, y_1$ and receives an encryption $C$ of $\mathcal{y}_\beta$ for $\beta \leftarrow \{0,1\}$ sampled by the challenger. The adversary is not allowed to make key queries after the challenge phase. When $A$ halts, it outputs $\beta' \in \{0,1\}$ and $S_0$ is the event that $\beta' = \beta$. Note that for any vector $x$ for which $A$ makes a secret key query, we must have $(x, y_0) \equiv (x, y_1) \bmod p$ if $A$ is a legitimate adversary.

**Game 1:** We modify the generation of $x$ and $\bar{x}$ for all secret key queries. Namely, instead of choosing $x \leftarrow \mathbb{Z}_p^2$ and then sampling $\bar{x}$, the challenger first chooses $\bar{x} = (\bar{x}_1, \ldots, \bar{x}_{2\ell})$ as $\bar{x}_i \leftarrow D_{\mathcal{Z},\sigma_1}$ for $i \in [\ell]$ and $\bar{x}_i \leftarrow D_{\mathcal{Z},\sigma_2}$ for $i \in [\ell + 1, 2\ell]$ and then sets $x := \bar{x} \bmod p$. We claim that this changes the joint distribution of $(x, \bar{x})$ 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 $D_{\mathcal{Z},\sigma_1}$ for $i \in [\ell]$ and $D_{\mathcal{Z},\sigma_2}$ 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 $D_{\mathcal{Z},\sigma_1}$ or $D_{\mathcal{Z},\sigma_2}$. This follows from $\sigma_1, \sigma_2 \geq p \cdot \Omega(\sqrt{\lambda})$ and Lemma 1. Therefore, we have that $|\mathbb{Pr}[S_1] - \mathbb{Pr}[S_0]| \leq 2^{-\Omega(\lambda)}$.

**Game 2:** We modify the generation of $C = (c_0, c_1)$ in the challenge phase. Namely, at the outset of the game, the challenger picks $s \leftarrow \mathbb{Z}_q^n$, $e \leftarrow D_{\mathcal{Z},\alpha'q}$ (which may be chosen ahead of time) as well as $Z \leftarrow D_{\mathcal{Z},\sigma_0}^{2\ell \times m}$. Let $V \in \mathbb{Z}^{(m + 2\ell)\times m}$ be the matrix that is obtained by putting $I_m$ on top of $Z$, where $I_m$ is the unit matrix of size $m$. We then set the ciphertext $C = (c_0, c_1) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^{2\ell}$ as

$$b = \mathbf{As} + e \quad (c_0 || c_1) = \text{ReRand}(V, b, \alpha'q, \alpha/(2\alpha')) + p^{k-1} \cdot \mathcal{y}_\beta$$

(4)

where ReRand is from Lemma 2. We claim that this change alters the view of the adversary only negligibly. To show this, we first observe that $s_1(V) \leq \sqrt{1 + s_1(Z)^2} \leq O(\sigma_0 \sqrt{m})$ holds with all but $2^{-\Omega(m)} \leq 2^{-\Omega(\lambda)}$ probability by Lemma 4. By Lemma 2 and our parameter choices, we have

$$c_0 = I_m \cdot \mathbf{As} + e_0 = \mathbf{As} + e_0,$$

$$c_1 = Z \cdot \mathbf{As} + e_1 + p^{k-1} \cdot \mathcal{y}_\beta = Us + e_1 + p^{k-1} \cdot \mathcal{y}_\beta,$$

where $e_0$ and $e_1$ are within statistical distance $2^{-\Omega(\lambda)}$ from $D_{\mathcal{Z},\alpha'q}^m$. Therefore, we have that $|\mathbb{Pr}[S_2] - \mathbb{Pr}[S_1]| \leq 2^{-\Omega(\lambda)}$.

**Game 3:** We further modify the generation of $C = (c_0, c_1)$ in the challenge phase. Instead of setting $b = \mathbf{As} + e$, we choose $b = \mathbf{u} + e$, where $\mathbf{u} \leftarrow \mathbb{Z}_q^n$. Then, the ciphertext is set as in Equation (4). Under the LWE assumption, we have that $|\mathbb{Pr}[S_3] - \mathbb{Pr}[S_2]|$ is negligible.

**Game 4:** We modify the generation of $C = (c_0, c_1)$ once more. Namely, the ciphertext is now set as

$$c_0 = \mathbf{u} + e_0, \quad c_1 = Z \cdot \mathbf{u} + e_1 + p^{k-1} \cdot \mathcal{y}_\beta,$$

where $\mathbf{u} \leftarrow \mathbb{Z}_q^n$ and $e_0, e_1 \leftarrow D_{\mathcal{Z},\sigma_0}^m$. Similarly to Game 2, this change does not alter the view of the adversary much. By Lemma 2 and our parameter choices, we have that $|\mathbb{Pr}[S_4] - \mathbb{Pr}[S_3]| \leq 2^{-\Omega(\lambda)}$. Below, we prove that $\mathbb{Pr}[S_4]$ is exponentially close to $1/2$, which will complete the proof.
Define $y = y_1 - y_0 \in \mathbb{Z}_p^{2\ell}$. Let \( \{x_{ij} \in \mathbb{Z}_p^{2\ell}\}_{j \in [L']} \) be the vectors corresponding to the secret key queries made by $A$, where $L' \leq L$. As $A$ is a legitimate adversary, we have \( (x_{ij}, y) = 0 \mod p \) for each secret key query $x_{ij}$. The view of the adversary contains $L'$ tuples \( \{x_{ij}, x_{ij}, z_{ij}\}_{j \in [L']} \), where the vectors \( \{x_{ij}\}_{j \in [L']} \) form a $\mathbb{Z}_p$-basis of a subspace of the $(2\ell - 1)$-dimensional vector space $y \perp := \{x \in \mathbb{Z}_p^{2\ell} : (x, y) = 0 \mod p\}$. We define $X_{\text{top}} \in \mathbb{Z}^{L' \times 2\ell}$ as the matrix whose $j$-th row is $x_{ij}'$ for $j \in [L']$.

We say that $X_{\text{top}} \in \mathbb{Z}^{L' \times 2\ell}$ is good when we can choose $U \in \mathbb{Z}^{2\ell \times 2\ell}$ such that \( |\det U| = 1 \), $X_{\text{top}} \cdot U = (I_{L'} | 0)$, and every row of $U$ has norm $\leq O(\sqrt{n\ell \sigma_2})$ (see Lemma 5). For a good $X_{\text{top}}$, we can define $X \in \mathbb{Z}^{2\ell \times 2\ell}$ as $X := U^{-1}$. It can be seen that the upper $L'$ rows of $X$ corresponds to $X_{\text{top}}$. We denote the lower $2\ell - L'$ rows of the matrix as $X_{\text{bot}}$. We note that since $X$ is invertible over $\mathbb{Z}$, so is it modulo $q$. Without loss of generality, we assume that $U$ and $X_{\text{bot}}$ are deterministically determined from $X_{\text{top}}$. (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 $U$ and $X_{\text{bot}}$ from $X_{\text{top}}$. This does not cause any problem in our proof because all the following arguments are information theoretic.

We state the following lemmas:

**Lemma 10** Assume that $\sigma_0 \geq \Omega(pk\ell \sigma_2)$ and $q^{n+1}/p^m \leq 2^{\Omega(\kappa)}$. Then the following distributions are within $2^{-\Omega(\kappa)}$ statistical distance:

\[
(A, u, ZA, X_{\text{top}}Z, X_{\text{bot}}Zu) \approx (A, u, ZA, X_{\text{top}}Z, v)
\]

where $A \leftarrow \mathbb{Z}_p^{m \times n}$, $u \leftarrow \mathbb{Z}_p^m$, $Z \leftarrow D_{\mathbb{Z}_p^{m \times m}}^{2\ell \times m}$, each row of $X_{\text{top}} \in \mathbb{Z}^{L' \times 2\ell}$ is chosen from $D_{\mathbb{Z}_p^{m \times m}}^{2\ell} \times D_{\mathbb{Z}_p^{m \times m}}^{\ell}$, and $v \leftarrow \mathbb{Z}_p^{2\ell - L'}$. Note that if $X_{\text{bot}}$ is not good, then $X_{\text{bot}}$ is undefined. In such a case, the term $X_{\text{bot}}Zu$ is replaced with $\bot$.

**Lemma 11** If there exists an adversary $A$ whose advantage in Game 4 is $\varepsilon$, then there exists another (unbounded) adversary $B$ whose distinguishing advantage between the two distributions in Lemma 10 is $\varepsilon/Q^{L'}$.

Given these two lemmas, we can conclude the proof of Theorem 5 since these imply $\varepsilon/Q^{L'} < 2^{-\Omega(\kappa)}$ and thus

\[
\varepsilon < Q^{L} \cdot 2^{-\Omega(\kappa)} = 2^{\Omega(L \log \lambda) - \Omega(\kappa)} \leq 2^{-\Omega(\lambda)}.
\]

It remains to prove Lemmas 10 and 11.

**Proof:** [Proof of Lemma 10] By Lemma 5, matrix $X_{\text{top}}$ is good with all but $2^{-\Omega(\kappa)}$ probability. In the following, let us fix good $X_{\text{top}}$ and prove that the above two distributions are $2^{-\Omega(\kappa)}$-close. We first consider the distribution $X_{\text{bot}}Z$ conditioned on $X_{\text{top}}Z$. Note that in $X_{\text{top}}Z$ and $X_{\text{bot}}Z$, matrices $X_{\text{top}}$ and $X_{\text{bot}}$ act in parallel on the columns of $Z$. We can hence restrict ourselves to the distribution of $X_{\text{bot}}Z_i$, conditioned on $X_{\text{top}}Z_i$, with $z_i$ sampled from $D_{\mathbb{Z}_p^{m \times m}}^{2\ell \times m}$. Let $b_i = X_{\text{top}}Z_i \in \mathbb{Z}^{L'}$ and fix $z_i^* \in \mathbb{Z}^{2\ell}$ arbitrary such that $b_i = X_{\text{top}}Z_i^*$. The distribution of $z_i$ given $(X_{\text{top}}Z_i^*)$ is $D_{\mathbb{Z}_p^{m \times m}}^{2\ell \times m}$, with $\Lambda = \{x \in \mathbb{Z}^{2\ell} : X_{\text{top}}Z_i = 0\}$. Therefore, we have that given $X_{\text{top}}Z_i$, the vector $X_{\text{bot}}Z_i$ is distributed as $X_{\text{bot}} \cdot D_{\mathbb{Z}_p^{m \times m}}^{2\ell \times m}$.

Let $U_{\text{lef}}$ (resp. $U_{\text{rig}}$) denote the left $L'$ (resp. right $2\ell - L'$) columns of $U$. We are to show that the distribution $X_{\text{bot}} \cdot D_{\mathbb{Z}_p^{m \times m}}^{2\ell \times m}$ is $D_{\mathbb{Z}_p^{m \times m}}^{2\ell \times m} \cdot \Sigma^{-1} \cdot w$, where $\Sigma = U_{\text{rig}} U_{\text{rig}}$ and $w = -\sqrt{\Sigma} U_{\text{rig}} U_{\text{lef}} b_i$. To see this, we first show that the supports of both distributions are the same. More specifically, we prove $X_{\text{bot}} \cdot \Lambda = \mathbb{Z}^{2\ell - L'}$. To do so, it suffices to show that for any $a \in \mathbb{Z}^{2\ell - L'}$, we have $a \in X_{\text{bot}} \cdot \Lambda$. By the construction of $U$, we have $X_{\text{top}} U_{\text{rig}} = 0$ and $X_{\text{bot}} U_{\text{rig}} = I_{2\ell - L'}$. Now, $a \in X_{\text{bot}} \cdot \Lambda$ follows because we have $a = X_{\text{bot}} \cdot (U_{\text{rig}} a)$ and $X_{\text{top}} \cdot U_{\text{rig}} a = 0$. We next evaluate the probability of $a \in \mathbb{Z}^{2\ell - L'}$ being output by $X_{\text{bot}} \cdot D_{\mathbb{Z}_p^{m \times m}}^{2\ell \times m}$ for $a'$ that is the unique vector in $\Lambda + z_i^*$ satisfying $a = X_{\text{bot}} \cdot a'$.
Since $\mathbf{a}' \in \Lambda + \mathbf{z}_i^*$, we have $\mathbf{X}_{top} \cdot \mathbf{a}' = \mathbf{X}_{top}(\mathbf{a}' - \mathbf{z}_i^*) + \mathbf{X}_{top} \mathbf{z}_i^* = 0 + \mathbf{b}_i = \mathbf{b}_i$. Therefore, the vector $\mathbf{a}'$ can be written as $\mathbf{a}' = \mathbf{X}^{-1}(\mathbf{b}_i \| \mathbf{a}) = \mathbf{U}(\mathbf{b}_i \| \mathbf{a}) = \mathbf{U}_{top} \mathbf{b}_i + \mathbf{U}_{rig} \mathbf{a}$. The probability we consider is proportional to

$$
\exp\left(-\pi \|\mathbf{a}'\|^2/\sigma_0^2\right)
= \exp\left(-\pi \|\mathbf{U}_{top} \mathbf{b}_i + \mathbf{U}_{rig} \mathbf{a}\|^2/\sigma_0^2\right)
= \exp\left(-\pi \|\sqrt{\mathbf{U}} \mathbf{a} + \sqrt{\mathbf{U}}^{-1} \mathbf{U}_{rig} \mathbf{U}_{top} \mathbf{b}_i\|^2/\sigma_0^2\right)
\cdot \exp\left(-\pi (\|\mathbf{U}_{top} \mathbf{b}_i\|^2 - \|\sqrt{\mathbf{U}}^{-1} \mathbf{U}_{rig} \mathbf{U}_{top} \mathbf{b}_i\|^2)/\sigma_0^2\right)
\text{ does not depend on } \mathbf{a}
\propto \exp\left(-\pi \|\sqrt{\mathbf{U}}(\mathbf{a} - \mathbf{w})\|^2/\sigma_0^2\right).
$$

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

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 $\mathbf{XZ} \mathbf{A}$ instead of $\mathbf{ZA}$. We do not lose 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 $\mathbf{X}$ from the left. Furthermore, we observe that $\mathbf{XZ} \mathbf{A}$ is the vertical concatenation of $\mathbf{X}_{top} \mathbf{Z} \mathbf{A}$ and $\mathbf{X}_{bot} \mathbf{Z} \mathbf{A}$. 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 $\mathbf{Y} := \mathbf{X}_{bot} \mathbf{Z} \sim (D_{\mathbb{Z}^{2\ell-L'},\sigma_0.\sqrt{\mathbf{U}}^{-1}.\mathbf{w}})^m$. To complete the proof, we will show that the following distributions are statistically close:

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

where $\mathbf{B} \leftarrow \mathbb{Z}_q^{(2\ell-L') \times n}$, and $\mathbf{v} \leftarrow \mathbb{Z}_q^{2\ell-L'}$. We first show that $(\mathbf{Y} \mathbf{u} \mod p)$ is within $2^{\Omega(-\lambda)}$ statistical distance from the uniform distribution over $\mathbb{Z}_p^{2\ell-L'}$. This follows by setting $\Lambda = \mathbb{Z}^{2\ell-L'}$ and $\Lambda = p \cdot \mathbb{Z}^{2\ell-L'}$ and applying Lemma 1 to $\mathbf{Y}$ in a column-wise manner. We check that the parameters satisfy the required condition of Lemma 1. We have

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

$$\geq \Omega((a^1/2 \ell \sigma_2)^{-1}),$$

where the last inequality follows from the upper bound on the norms of the rows of $\mathbf{U}$. We therefore have $\sigma_0^{-s_{2\ell-L'}(\sqrt{\mathbf{U}}^{-1})} \geq p \cdot \Omega(\sqrt{\pi})$ by our choice of $\sigma_0$. We then finally apply Lemma 3 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 $q^{n+1}/p^m \leq 2^{-\Omega(n)}$. This completes the proof of Lemma 10.

**Proof:** [Proof of Lemma 11] 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 $\{i_j\}_{j \in \mathcal{L}'} \in \mathcal{Q}^{L'}$ for which the adversary makes key queries. The public key $\mathbf{pk} = (\mathbf{A}, \mathbf{U} = \mathbf{Z} \cdot \mathbf{A})$ and the master key $\mathbf{msk} = \mathbf{Z}$ 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 $\{\mathbf{x}_i\}_{i \in \mathcal{Q}}$ as follows. When $i \in \{i_j\}_{j \in \mathcal{L}'}$, there exists $j$ such that $i = i_j$. Then algorithm $\mathcal{B}$ sets $\hat{\mathbf{x}}_i^1$ to be the $j$-th row of the given matrix $\mathbf{X}_{top}$. Otherwise, it chooses $\hat{\mathbf{x}}_i$ as in Game 4. The key queries are handled as follows. Whenever $\mathcal{A}$ queries key for $\mathbf{x}_i$ such that $i \notin \{i_j\}_{j \in \mathcal{L}'}$, 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

$$c_0 = \mathbf{u}, \quad c_1 = \mathbf{X}^{-1} \cdot (\mathbf{X}_{top} \mathbf{Z} \mathbf{u} \| \mathbf{v}) + \mathbf{e}_1 + y_\beta.$$
We can observe that when \( \mathbf{v} = \mathbf{X}_{\text{bot}}\mathbf{Z}_{\text{u}} \), we have \( \mathbf{c}_1 = \mathbf{Z}_{\text{u}} + \mathbf{e}_1 + \mathbf{y}_\beta \) 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 \( \mathbf{X} \cdot \mathbf{c}_1 \mod q \) is independent of \( \beta \). 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 \( \beta \) as well (recall that \( \mathbf{X} \) is information theoretically known to \( \mathcal{A} \), which means that, if \( \mathbf{c}_1 \) carries any information on \( \beta \), so does \( \mathbf{X} \cdot \mathbf{c}_1 \mod q \). The first \( L' \) entries of \( \mathbf{X} \cdot \mathbf{c}_1 \) (namely, \( \mathbf{X}_{\text{top}} \cdot \mathbf{c}_1 \)) do not depend on \( \beta \) because we have the equality \( p^{k-1} \cdot \mathbf{X}_{\text{top}} \cdot \mathbf{y}_0 = p^{k-1} \cdot \mathbf{X}_{\text{top}} \cdot \mathbf{y}_1 \mod q \), by construction of \( \mathbf{X}_{\text{top}} \). 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 \( \mathbf{v} = \mathbf{X}_{\text{bot}}\mathbf{Z}_{\text{u}} \) and a game that is independent of \( \beta \) 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 11. \( \square \)

## 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 meet the requirements for our generic Trace-and-Revoke construction, 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. Abdalla et al. (2015), two DDH-based adaptively secure IPFEs modulo the group size \( q \) have been proposed Agrawal et al. (2016); Benhamouda et al. (2017). 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 \( m \cdot \langle \mathbf{x}_{\text{id}}, \mathbf{v}_R \rangle \) 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 Agrawal et al. (2016) by removing the \( \langle \mathbf{x}_{\text{id}}, \mathbf{v}_R \rangle \) component before taking the discrete logarithm.

- **Setup**\((1^\lambda, 1^t, 1^r, L)\). Choose a cyclic group \( \mathbb{G} \) of prime order \( q \) along with two generators \( g, h \leftarrow \mathbb{G} \). DDH in \( \mathbb{G} \) should be \( 2^\lambda \)-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 \( s_i, t_i \leftarrow \mathbb{Z}_q \) and compute \( h_i = g^{s_i} \cdot h^{t_i} \). Define

\[
\text{msk} := (s, t) \quad \text{and} \quad \text{pk} := (\mathbb{G}, g, h, \{h_i\}_{i \in [\ell]}).
\]

- **KeyGen**\((\text{msk}, \text{id})\). Sample \( \mathbf{x}_{\text{id}} \leftarrow \mathbb{Z}_q^\ell \). Set \( \text{sk}_{\text{id}} = (\langle \mathbf{x}_{\text{id}}, s \rangle, \langle \mathbf{x}_{\text{id}}, t \rangle) \in \mathbb{Z}_q^{2\ell} \) and \( \text{pk}_{\text{id}} = \mathbf{x}_{\text{id}} \).

- **Enc**\((\text{pk}, \mathcal{R}, m)\) proceeds as follows to encrypt \( m \in \{1, \ldots, L\} \):
  1. Compute \( \mathbf{v}_R \in \mathbb{Z}_q^\ell \setminus \{\mathbf{0}\} \) such that \( \langle \mathbf{x}_{\text{id}}, \mathbf{v}_R \rangle = 0 \) for every \( \text{id} \in \mathcal{R} \).
  2. Set \( \mathbf{y} = m \cdot \mathbf{v}_R \) and sample \( r \leftarrow \mathbb{Z}_q \).
  3. Compute \( D_0 = g^r, D_1 = h^r \) and \( E_i = g^{s_i} \cdot h_i^r \) for all \( i \leq \ell \).

The ciphertext \( C \) is \((D_0, D_1, E_1, \ldots, E_\ell, \mathcal{R})\).
• Dec(sk_{id}, C). Write $C = (D_0, D_1, E_1, \ldots, E_\ell, \mathcal{R})$ and let $x_{id}$ denote the vector corresponding to $sk_{id} = (s_x, t_x)$. Compute:

$$C_{x_{id}} = \left( \prod_{i=1}^{\ell} E_i^{x_{id,i}} \right)/(D_0^{s_x} \cdot D_1^{t_x}).$$

Then, recover $v_{\mathcal{R}}$ from $\mathcal{R}$ and output the base-$g$ logarithm of $C_{x_{id}}^{-1/(v_{\mathcal{R}}, x_{id})}$.

• Trace(pd, $S$, $\mathcal{R}, O^D$) 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 $x_{id}$ is replaced by an inner product, which can be efficiently computed.

**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 $sk_{id}$ 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 Agrawal et al. (2016), Agrawal et al. described two IPFEs relying on the algebraic framework of Paillier’s encryption scheme Paillier (1999). 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 how to instantiate our trace-and-revoke construction using this IPFE over $\mathbb{Z}^\ell_N$.

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 Agrawal et al. (2016) 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.

• Setup(1^\lambda, 1^t, 1^r). Choose safe prime numbers $p = 2p' + 1$, $q = 2q' + 1$ with sufficiently large primes $p', q' > 2^{\text{Poly}(\lambda)}$, and compute $N = pq$. Then, sample $g' \leftarrow \mathbb{Z}_N^{\times}$ and compute $g = g'^{2N}$ mod $N^2$, which generates the subgroup of $(2N)$-th residues in $\mathbb{Z}_N^{\times}$ with overwhelming probability. Set $\ell = t + r + 1$ and sample $s$ from the integer Gaussian distribution $\mathcal{D}_{2\sigma^2}$ with standard deviation parameter $\sigma$ satisfying $\sigma \geq \sqrt{\ell}N^{\text{Poly}(\lambda)}$. Compute $h_i = g^s$ mod $N^2$ for all $i \leq \ell$. Define

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

• KeyGen(msk, id). Sample $x_{id} \in \mathbb{Z}^\ell$ with coefficients i.i.d. uniform in $\{0, \ldots, N - 1\}$. Set $sk_{id} = (x_{id}, s) \in \mathbb{Z}$ and $p_{id} = x_{id}$.

• Enc(pk, $\mathcal{R}$, $m$) proceeds as follows to encrypt $m \in \mathbb{Z}_N \setminus \{0\}$.

  1. Compute $v_{\mathcal{R}} \in \mathbb{Z}_p^\ell \setminus \{0\}$ such that $\langle x_{id}, v_{\mathcal{R}} \rangle = 0$ for every $id \in \mathcal{R}$. 25
2. Set \( y = m \cdot v_R \) and sample \( r \leftarrow \{0, \ldots, \lfloor N/4 \rfloor \} \).

3. Compute \( C_0 = g^r \mod N^2 \) and \( C_i = (1 + y_i N) \cdot h^r_i \mod N^2 \) for all \( i \leq \ell \).

The ciphertext \( C \) is \((C_0, C_1, \ldots, C_\ell, R)\).

- \( \text{Dec}(sk_{id}, C) \). Write \( C = (C_0, C_1, \ldots, C_\ell, R) \) and let \( x_{id} \) denote the vector corresponding to \( sk_{id} \). Compute:

\[
C_{x_{id}} = C_0^{-sk_{id}} \cdot \left( \prod_{i=1}^{\ell} C_i^{x_{id,i}} \right) \mod N^2.
\]

Then, recover \( v_R \) from \( R \) and output

\[
\left( \frac{C_{x_{id}} - 1 \mod N^2}{N} \right) (v_R, x_{id}).
\]

- \( \text{Trace}(pd, S, R, O^D) \) 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 Agrawal et al. (2016) exactly as in Subsection 4.2. In more detail, the proof of Theorem 5 (Appendix F) in Agrawal et al. (2016) can be modified to show that the advantage of the adversary in Game 3 is negligible even when \( \sigma \) 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 \( \text{id} \) corresponds a secret key \( sk_{id} \) 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 \( Z_N \setminus \{0\} \) to a ciphertext of bit-length \( O((t + r) \log N) \). To compensate for the number field sieve, we must choose \( \log N = \tilde{\Omega}(\lambda^3) \).

**Acknowledgements:** We thank Benoît Libert for helpful discussions. The authors would also like to thank the anonymous referees for their valuable comments and helpful suggestions. 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 JPMJCR1668, Japan.

**References**


6 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 Boneh and Waters (2006). 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 Boneh and Waters (2006). 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 Naor and Pinkas (2000); Dodis and Fazio (2003), a probe ciphertext may be distinguished from a normal ciphertext using a revoked key. In the polynomial interpolation based method in Naor and Pinkas (2000), 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 Naor and Pinkas (2000) 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 Chor et al. (1994); Naor and Pinkas (2000); Stinson and Wei (1998a,b); Naor et al. (2001); Ngo et al. (2013), 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 Nishimaki et al. (2016) may be instantiated with the bounded collusion functional encryption scheme from Gorbunov et al. (2012). 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 $O((r + t)^5 Poly(\lambda))$. If the compiler is instantiated with the bounded collusion scheme of Goldwasser et al. (2013) (compiled with Gorbunov et al. (2012)), then the ciphertext size still grows as $O((r + t)^5 Poly(\lambda))$, 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 $\text{Poly}(\lambda)$ factors above are unspecified, and possibly large: for instance, the circuit in Gorbunov et al. (2012) is represented using randomizing polynomials which adds a polynomial factor blow-up. Similarly, using the bounded collusion FE of Agrawal and Rosen (2016) leads to better asymptotic bounds $O(r + t)^3\text{Poly}(\lambda)$ 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 Agrawal and Rosen (2016) and an additional factor $(r + t)$ is incurred due to circuit size dependence.