Fully Collusion Resistant Trace-and-Revoke Functional Encryption for Arbitrary Identities

Fucai Luo[1], Saif Al-Kuwari[2], Haiyan Wang[1], and Xingfu Yan[3]

[1] Department of New Networks, Peng Cheng Laboratory, Shenzhen, China
lfucai@126.com
[2] College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar
[3] School of Computer Science, South China Normal University, Guangzhou, China

Abstract. Functional Encryption (FE) has been extensively studied in the recent years, mainly focusing on the feasibility of constructing FE for general functionalities, as well as some realizations for restricted functionalities of practical interest, such as inner-product. However, little consideration has been given to the issue of key leakage on FE. The property of FE that allows multiple users to obtain the same functional keys from the holder of the master secret key raises an important problem: if some users leak their keys or collude to create a pirated decoder, how can we identify at least one of those users, given some information about the compromised keys or the pirated decoder? Moreover, how do we disable the decryption capabilities of those users (i.e. traitors)?

Two recent works have offered potential solutions to the above traitor scenario. However, the two solutions satisfy weaker notions of security and traceability, can only tolerate bounded collusions (i.e., there is an a priori bound on the number of keys the pirated decoder obtains), or can only handle a polynomially large universe of possible identities. In this paper, we study trace-and-revoke mechanism on FE and provide the first construction of trace-and-revoke FE that supports arbitrary identities, is both fully collusion resistant and fully anonymous. Our construction relies on a generic transformation from revocable predicate functional encryption with broadcast (RPFE with broadcast, which is an extension of revocable predicate encryption with broadcast proposed by Kim and J. Wu at ASIACRYPT'2020) to trace-and-revoke FE. Since this construction admits a generic construction of trace-and-revoke inner-product FE (IPFE), we instantiate the trace-and-revoke IPFE from the well-studied Learning with Errors (LWE). This is achieved by proposing a new LWE-based attribute-based IPFE (ABIPFE) scheme to instantiate RPFE with broadcast.

Keywords: Functional encryption, Trace-and-revoke system, Inner-product functional encryption, Traceability.

1 Introduction

Functional encryption (FE) [14,53] facilitates fine-grained access control over encrypted data, which overcomes the all-or-nothing property of public key encryption (PKE). In FE, the holder of the master secret key is able to distribute arbitrary functional (decryption) keys that allow recipients of the keys to learn specific functions of the encrypted data, and nothing else. That is, given an encryption of a message x and a secret key sk_f for a function f, the holder of sk_f can only learn the value $f(x)$. However, such multi-receiver encryption system raises an important problem: since FE allows multiple users to obtain the same functional keys from the holder of the master secret key, what if some of these functional keys are leaked or a pirated decoder (capable of decrypting the ciphertext with a non-negligible probability) is found? how can we identify the source of the key leakage? These situations makes it important to construct an efficient traitor tracing mechanism on FE that can identify such malicious user(s). In this paper, we provide affirmative answers to these questions.

Traitor tracing mechanisms on FE were only proposed recently [23, 43]. In [23], the authors introduced the notion of traceable FE and gave the first construction of traceable functional encryption for inner products (a.k.a. inner-product functional encryption (IPFE)). In additional to constructing a more efficient traceable IPFE scheme (compared with [23]), the authors in [43] also considered how to dismiss identified traitor(s) by establishing a revocation mechanism compatible with the underlying traitor tracing mechanism (called "trace-and-revoke IPFE"). Nevertheless, we observe some shortcomings on [23,43]. First, on the security side, the scheme of [23] is proven selectively secure against chosen-plaintext attacks under the Decisional Bilinear Diffie-Hellman (DBDH) assumption [25] and supports a weaker notion of black-box traceability that they called one-target black-box traceability (see Remark 2) under the Decisional Diffie-Hellman (DDH) assumption [10]. Second, on the size of identities supported by the system, both schemes in [23] and [43] can only handle a polynomial-size identity space, which in turn led to their schemes only tolerating bounded collusions[1]. Moreover, in [23,43], the supported identities are represented by a set $[N] = \{1, \ldots, N\}$ and each identity (i.e. user) is assigned and identified by a unique index $i \in [N]$. This means that the tracing authority needs to keep track of the users who have been issued decryption keys (using the issued indices) and thus it needs to separately maintain an explicit mapping (as a look-up table) between an index $i \in [N]$ and the user identification information. This prevents their schemes from being used in the anonymous setting and violates a desired privacy property, where honest users are no longer fully anonymous; fully anonymous means that the tracing authority does not need to separately record the user information to index mapping, and honest users can even remain anonymous to the tracing authority and the key issuer [2] [48].

These motivate us to consider the following question:

Is it possible to construct a trace-and-revoke (inner-product) FE scheme that simultaneously supports arbitrary identities (i.e. its size is exponential), and is both fully collusion resistant and fully anonymous?

Inner-Product Functional Encryption. In an inner-product functional encryption system [1], an authority uses its master secret key msk to generate a secret key $sk_{\mathbf{y}}$ associated with a vector $\mathbf{y}$. Given a ciphertext $\mathbf{c}$ designed to encrypt a message $\mathbf{x}$, any user holding $sk_{\mathbf{y}}$ can decrypt $\mathbf{c}$ to obtain the inner-product $\langle \mathbf{y}, \mathbf{x} \rangle$, but not $\mathbf{x}$.

Most of the existing extensions of IPFE mainly focus on function-hiding [38, 58], multi-input [3,22], and (decentralized) multi-client [21,41]. However, for any IPFE scheme, even with function-hiding, the original plaintext $\mathbf{x}$ can be recovered if a sufficiently large amount of different secret keys is released [2]. The impact of this inherent security issue in IPFE can be drastic on some scenarios where the data owner uses the scheme to encrypt database and many users access that database; this issue can indeed be addressed by FE schemes for general functions based on indistinguishable obfuscation or multilinear maps [26, 27], but such FE schemes are usually too inefficient.

Attribute-Based (Inner-Product) Functional Encryption. Attribute-based functional encryption (ABFE) was initially introduced by Chen *et al.* [19]. ABFE aggregates the features of attribute-based encryption (ABE) and FE, so that if a user's attribute satisfies a specified policy

[1] A traceable system is t-collusion resistant if tracing works as long as the pirated decoder obtains fewer than t decryption keys, and the parameters may depend on t. We say that the system is fully collusion resistant if t is of arbitrary polynomial size.

[2] Instead of requesting a secret key for identity id by sending id to the key issuer directly, the user can obtain the secret key for id by running a multi-party computation (MPC) with the key issuer so that the key issuer learns nothing about id.

function (e.g. $f(x) = 0$, where f is the policy function and x is the attribute), then the user can obtain the desired function value by decrypting the corresponding ciphertext.

In [19], the authors presented an attribute-based inner-product functional encryption (ABIPFE) that supports monotone span programs based on decisional assumptions on bilinear groups, building on the ABE framework proposed in [60]. However, the scheme in [19] can only satisfy weak security, where the adversary is not allowed to make any key query that can decrypt the challenge ciphertext, which weakens the standard security definition of IPFE. Based on any "dual" ABE, Abdalla *et al.* [2] proposed two ABIPFE schemes based on SXDH assumption, which supports monotone span programs. Moreover, Abdalla *et al.* proposed two lattice-based identity-based IPFE schemes by combining existing lattice-based IBE schemes with the IPFE scheme of Agrawal *et al.* [6]. Since the lattice-based ABE scheme for circuits of [12] is obtained by generalizing the puncturing technique used in the security proof of existing lattice-based identity-based encryption (IBE) schemes (e.g. [5]), Abdalla *et al.* analyzed the security proof of their lattice-based identity-based IPFE schemes and pointed out some technical limitations in the security proof strategy which prevent them from extending their schemes to the ABE case. Consequently, they left combining ABE for circuits with IPFE as an open problem.

Recently, by combining lattice-based ABE scheme of [12] with the IPFE schemes of Agrawal *et al.* [6], two different LWE-based ABIPFE schemes were proposed by Luo and Al-Kuwari [42] and Pal and Dutta [49], respectively. Unsurprisingly, both schemes suffer from the above-mentioned technical limitations in the security proofs. Concurrently, Lai *et al.* [40] proposed a new two-stage sampling technique that turns out to be useful in improving security and efficiency of the simulation-based functional encryption. In particular, they proposed an LWE-based ABIPFE scheme that circumvents the above technical limitations using the two-stage sampling technique in the security proof. However, their LWE-based ABIPFE scheme fails to satisfy the standard selective security of ABIPFE, which we discuss in Remark 1. In this work, we propose a new, more efficient LWE-based ABIPFE scheme that satisfies the standard selective security of ABIPFE.

1.1 Our Contribution

In this paper, we address the trace-and-revoke scenario by proposing a generic construction of trace-and-revoke (inner-product) functional encryption scheme that is able to support arbitrary identities, and is both fully collusion resistant and fully anonymous. Furthermore, we instantiate our generic construction of trace-and-revoke inner-product functional encryption scheme from Learning with Errors (LWE) assumption [52]; LWE has a simple algebraic structure and is widely believed to be quantum-resistant. Our results can be summarized as follows:

1. We introduce a new primitive called (secret-key) revocable predicate functional encryption (RPFE) with broadcast (cf. Definition 9), which extends the previous notion (including the corresponding security notions) of (secret-key) revocable predicate encryption (RPE) with broadcast [39].
2. We give a generic construction of secret-key RPFE with broadcast by extending the construction of secret-key RPE with broadcast of [39]; see Section 4.2. In addition, we show that it is straightforward to obtain a generic construction of secret-key revocable predicate inner-product functional encryption (RPIPFE) with broadcast by simply replacing the underlying ABFE with ABIPFE; see Section 4.3.
3. Using the above secret-key RPFE (resp. RPIPFE) with broadcast, we provide a generic construction of trace-and-revoke FE (resp. IPFE), which extends the identity-based trace-and-revoke scheme of [39]. As an extension of [39], our construction naturally inherits some useful

properties the scheme of [39] achieves, such as supporting arbitrary identities, full collusion resistance, and full anonymity.

4. We instantiate our construction of trace-and-revoke IPFE with the LWE-based secret-key RPFE with broadcast (cf. Section 5.3). To achieve this, we first instantiate our construction of secret-key RPFE with broadcast using a new LWE-based ABIPFE scheme that we propose in Section 3.

In addition, we compare our trace-and-revoke IPFE scheme with the previous work [23, 43]; a brief comparison is provided in Table 1. Note that we cannot directly instantiate the secret-key RPFE with broadcast with the previous LWE-based ABIPFE schemes [40,42,49], as these schemes do not satisfy adaptive security (see Section 3 for more detail).

We remark that our generic construction of trace-and-revoke FE provides some possible instantiations of trace-and-revoke FE from standard assumptions if attribute-based FE schemes are constructed from standard assumptions (there is no evidence at the moment that this is not possible), or maybe we can obtain trace-and-revoke FE for other restricted functionalities of practical interest (e.g. quadratic functions [8]) from our trace-and-revoke FE framework. This provides a significant improvement over the previous work [23, 43] that restricts the functionality to inner-product functions.

Table 1: Comparison with Previous Work.

Scheme	Identity Space	Collusion Resistance	Full Anonymity
[23]	Polynomial-size	Bounded	No
[43]	Polynomial-size	Bounded	No
Ours	Exponential-size	Full	Yes

1.2 Construction Overview

Since our work can be seen as an extension to [39], we start with a brief overview of the approach of [48] and the identity-based trace-and-revoke scheme of [39]. Then, we show how their work can be extended to build trace-and-revoke (inner-product) functional encryption scheme.

The Approach of [48]. In [48], Nishimaki *et al.* abstracted the tracing problem as an "oracle jump-finding" problem and proposed a general tracing algorithm for private linear broadcast encryption (PLBE) that supports an exponential identity space. The PLBE scheme, initially introduced by Boneh *et al.* [13] for constructing a traitor tracing scheme, is a multi-receiver encryption scheme, where the decryption key is associated with an index $i \in [N]$ and the ciphertext is associated with a message m and a secret index $j \in [N]$. The correctness property guarantees that a decryption key sk_i for index i can decrypt all ciphertexts encrypted to indices j where $i \leq j$. There are two different algorithms to generate ciphertexts: 1) the public encryption algorithm, which allows anyone to encrypt to the index N, and the resulting ciphertexts can be decrypted by secret keys sk_i for all $i \in [N]$; 2) the secret encryption algorithm, which allows the tracing authority who holds a tracing key to encrypt to indices $j \leq [N]$. The two encryption algorithms and some requirements (i.e. "index-hiding" and "message-hiding" [13]) that PLBE needs to meet are the crux of building a traitor tracing scheme from PLBE.

The traitor tracing scheme of [48] relies on a PLBE scheme that satisfies the following more general notion of "index-hiding" security, which states that a ciphertext encrypted to index $k_1 \in [N]$ should be indistinguishable from a ciphertext encrypted to index $k_2 \in [N]$ as long as the adversary does not have any keys in the interval $(k_1, k_2]$. The authors defined a tracing algorithm that runs an efficient algorithm designed for a (N, q, δ, C)-noisy jump finding problem they introduced as a subroutine (see Definition 5 of [48]). Then, they reduced the correctness proof of their tracing algorithm to the (N, q, δ, C)-noisy jump finding problem. However, this construction is limited by the fact that the ciphertexts need to grow with the identity bit-length. Consequently, the authors introduced a generalization of PLBE that supports slightly more general broadcast sets and defined a new tracing algorithm relying on a $(N, r, q, \delta, \varepsilon)$-generalized jump-finding problem (cf. Definition 4).

Identity-Based Trace-and-Revoke Scheme [39]. Kim and J. Wu [39] proposed an identity-based trace-and-revoke scheme by combining an identity-based traitor tracing scheme based on the techniques developed in [30,48] with the combinatorial revocation scheme of [46]. Specifically, they first introduced a new primitive called (secret-key) predicate encryption with broadcast, which was inspired by the PLBE construction in [30]. In a secret-key predicate encryption with broadcast, the decryption key sk_x is associated with an attribute x, while the ciphertext is associated with a predicate f and a message m. The correctness property guarantees that a decryption key sk_x can decrypt all ciphertexts if $f(x) = 1$ and $\perp$ otherwise. Moreover, the policy f associated with a ciphertext is hidden irrespective of whether decryption succeeds or not. Here, "broadcast" refers to a public encryption algorithm that allows anyone to encrypt messages with respect to the "always-accept" policy (i.e. $f(x) = 1$ for all x); this allows anyone in the system to decrypt the resulting ciphertexts. From the previous work [13, 30], they noted that the secret-key predicate encryption with broadcast suffices to construct a fully collusion resistant traitor tracing scheme with short ciphertexts via the approach of [48]. To achieve that, they constructed a secret-key predicate encryption with broadcast by combining a mixed FE (for general circuits) and an ABE (for general circuits), where the public encryption (i.e. "broadcast") and secret encryption algorithms are built by combining that of the mixed FE (cf. Definition 5) with the ABE.

To further achieve revocation functionality, Kim and J. Wu [39] embedded the subset-cover set system of [46] into their proposed secret-key predicate encryption with broadcast, resulting in a secret-key revocable predicate encryption with broadcast. Finally, they showed how to directly build a fully collusion resistant trace-and-revoke scheme for arbitrary identities from the secret-key revocable predicate encryption with broadcast via [48].

Our Trace-and-Revoke (IP)FE Scheme. Recall that the main building blocks of the secret-key revocable predicate encryption with broadcast of [39] are a mixed FE and an ABE, where the ABE is used to encrypt messages and the function policy of the ABE is associated with an index and the mixed FE. That is, the scheme interweaves the mixed FE with the ABE by taking the attribute of the ABE as an input message of the mixed FE and the ciphertext of the mixed FE as an input of the function policy of the ABE.

Our main observation is that the interleaving of mixed FE and ABE is independent of the messages to be encrypted, and thus if we replace the underlying ABE scheme with an AB(IP)FE scheme while keeping the function policy unchanged, we can obtain a new primitive, which we call secret-key predicate functional encryption with broadcast. Our secret-key predicate functional encryption with broadcast that captures both "predicate functionality" and "function functionality" can be seen as an extension of the secret-key predicate encryption with broadcast proposed in [39]. The syntax of the secret-key predicate functional encryption with broadcast (see Definition 9) is the same as that of the secret-key predicate encryption with broadcast, except that the decryption key $sk_{x,g}$ is associated with an attribute x and a function g, and the decryption would recover a function value. In addition, we observe that the corresponding security requirements ("message hiding", "function hiding", and "broadcast security") that the secret-key predicate encryption with broadcast must satisfy can be easily generalized to the case of AB(IP)FE, and so are its corresponding security proofs, except for some security requirements of the (IP)FE, which we will point out in Section 4.

Like the secret-key predicate encryption with broadcast, our secret-key predicate functional encryption with broadcast requires the underlying AB(IP)FE scheme to satisfy adaptive security (cf. Definitions 7 and 8). However, to the best of our knowledge, no ABFE scheme from standard assumptions has yet been proposed. On the other hand, there are several ABIPFE schemes [40,42,49], but they all fail to achieve adaptive security via a standard complexity leveraging argument (cf. Section 3). This means that we cannot instantiate our secret-key revocable predicate functional encryption with broadcast from standard assumptions. Instead, we instantiate our secret-key revocable predicate inner-product functional encryption with broadcast from LWE by constructing an adaptively secure ABIPFE scheme based on LWE.

LWE-Based ABIPFE Scheme. At a high level, our LWE-based ABIPFE scheme is based on the LWE-based ABE scheme of Boneh *et al.* [12] and the LWE-based IPFE scheme of Agrawal *et al.* [6]. In the construction of our scheme, the generations of master public/secret keys are similar to that in [12]; in the key generation, we slightly twist the approach of [42,49] to insert the two-stage sampling technique of [40], which is similar to that of the identity-based IPFE scheme of [40]. In the selective security proof, we show that our LWE-based ABIPFE scheme satisfies the selective security in the standard model by combining the LWE proof technique of the LWE-based ABE scheme (i.e. programming the challenge attribute into the public parameters) with the two-stage sampling technique. Finally, we discuss how to upgrade our ABIPFE scheme from selective security to semi-adaptive security and adaptive security. We refer to Section 3 for details.

1.3 Related Work

In this section, we survey some of the related work on traitor tracing and trace-and-revoke schemes. Since we have already discussed the existing traceable IPFE scheme [23] and trace-and-revoke IPFE scheme [43], we omit them here to avoid redundancy.

Traitor tracing [20] is a multi-receiver encryption system that provides content distributors with a way to identify malicious receivers that build pirated decoders. Since its inception, a large body of work has been proposed based on a wide range of assumptions and settings. From the methodology perspective, most existing constructions can be roughly categorized into two main categories: combinatorial [9,20,24,34,46,55,56] and algebraic [4,15,30,32,33,36,39,47,48] approaches.

The general idea behind combinatorial constructions is to identify the traitors by analyzing keys that were carefully selected and placed in a pirated decoder. Chor, Fiat and Naor [20] proposed the first combinatorial traitor tracing scheme, which is either information theoretic security or is based on the security of any symmetric scheme of its choice. Similarly, Stinson and Wei [57] proposed the first combinatorial trace-and-revoke scheme, which combines broadcast encryption (act as a revocation mechanism) with the traitor tracing scheme.

On the other hand, the general idea behind algebraic schemes is to use some algebraic approaches to generate secret keys for users, and using some public-key techniques to perform the broadcasting. Boneh and Franklin [11] proposed the first algebraic traitor tracing scheme by applying error correcting codes to the discrete log representation problem. Later, Boneh and Waters [15] proposed the first algebraic trace-and-revoke scheme by constructing a new primitive called augmented broadcast encryption and showing that augmented broadcast encryption implies a trace-and-revoke scheme.

From the security and the supported identity space perspective, these traitor tracing and trace-and-revoke schemes can be broadly divided into the following categories: 1) schemes that are bounded collusion resistant, including but not limited to [4, 20, 47, 55]; 2) schemes that are fully collusion resistant, but can only handle a polynomial-size identity space, including but not limited to [13, 18, 30, 32]; 3) schemes that are fully collusion resistant and support an exponential identity space, including but not limited to [31, 33, 39, 48].

In addition, since traitor tracing and trace-and-revoke systems have also been studied in various settings, they can be classified according to the cryptographic primitives they target (e.g., broadcast encryption, identity-based encryption, attribute-based encryption, and functional encryption). In this work, we focus on trace-and-revoke functional encryption.

1.4 Organization

In Section 2, we provide an overview of the notations, cryptographic primitives, problems and algorithms we use in our constructions. We give a concrete construction of ABIPFE in Section 3. In Section 4, we first introduce a new cryptographic primitive that we call secret-key revocable predicate functional encryption with broadcast. Then we show how to construct a secret-key revocable predicate functional encryption with broadcast in a generic way, and instantiate the secret-key revocable predicate inner-product functional encryption with broadcast from LWE. We introduce the syntax of trace-and-revoke FE and provide a generic construction of trace-and-revoke FE in Section 5. Additionally, Section 5 also provides an instantiation of trace-and-revoke IPFE from LWE. Finally, Section 6 concludes the paper with final remarks and pointers to some existing open problems.

2 Preliminaries

Let PPT denote probabilistic polynomial-time. We say that a function is negligible, denoted $\text{negl}(n)$, if $\text{negl}(n)$ is asymptotically smaller than the inverse of any polynomial in n, and a function is overwhelming if it is $1 - negl(n)$. For simplicity, we let $[n] \triangleq \{1, \ldots, n\}$, and $[m, n]$ be the set of integers $\{m, m+1, \ldots, n\}$ for integers $1 \leq m \leq n$. The symbol $x \leftarrow \mathcal{D}$ indicates that x is sampled uniformly at random from the distribution or set $\mathcal{D}$. We use lower-case bold letter to denote vector $\mathbf{x}$ and upper-case bold letter to denote matrix $\mathbf{A}$. The transpose of $\mathbf{A}$ (resp. $\mathbf{x}$) is denoted as $\mathbf{A}^T$ (resp. $\mathbf{x}^T$). The column concatenation of matrices $\mathbf{A} \in \mathbb{Z}^{m \times k}$ and $\mathbf{B} \in \mathbb{Z}^{m \times k'}$ is represented by $(\mathbf{A}|\mathbf{B}) \in \mathbb{Z}^{m \times (k+k')}$.

We let r_i (resp. $\mathbf{x}_i$) denote the i-th component of any set r (resp. vector $\mathbf{x}$). We denote by $\langle \mathbf{v}, \mathbf{w} \rangle$ the inner-product of two vectors $\mathbf{v}, \mathbf{w}$. The ℓ_2 length of any vector $\mathbf{b} \in \mathbb{Z}^m$ is represented by $||\mathbf{b}||$, i.e. $||\mathbf{b}|| := \sqrt{\sum_{i=1}^{m} \mathbf{b}_i^2}$, and similarly, for any matrix $\mathbf{A} \in \mathbb{Z}^{m \times k}$, its matrix norm is denoted as $||\mathbf{A}||$, i.e. the ℓ_2 length of its longest column vector. In addition, the norm of its Gram-Schmidt (GS) orthogonalization is represented by $||\mathbf{A}||_{\text{GS}}$. We also define $||\mathbf{A}||_2 := \sup_{||\mathbf{e}||=1} ||\mathbf{A}\mathbf{e}||$ (known as the largest singular value). Then, it is not hard to prove that $||\mathbf{A}||_{\text{GS}} \leq ||\mathbf{A}|| \leq ||\mathbf{A}||_2 \leq \sqrt{m} \cdot ||\mathbf{A}||$ and $||\mathbf{A}\mathbf{B}||_2 \leq ||\mathbf{A}||_2 \cdot ||\mathbf{B}||_2$ for any $\mathbf{B} \in \mathbb{Z}^{k \times k'}$.

2.1 Lattices

In this section, we give a brief overview of Lattices, LWE and lattice trapdoors. For a more comprehensive discussion of these topics, refer to [7, 51, 52]. A q-ary integer lattice and a "shifted" coset are defined as follows:

Definition 1. *For $q \geq 2$, $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, and $\mathbf{u} \in \mathbb{Z}_q^n$, we define*

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

$$\Lambda_q^{\mathbf{u}}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \mathbf{u} \bmod q\}.$$

Note that if $\mathbf{y} \in \Lambda_q^{\mathbf{u}}(\mathbf{A})$, then $\Lambda_q^{\mathbf{u}}(\mathbf{A}) = \Lambda_q^{\perp}(\mathbf{A}) + \mathbf{y}$.

Discrete Gaussians. We let $\mathcal{L}$ denote a subset of $\mathbb{Z}^m$, the Gaussian function on $\mathbb{R}^m$ with the center $\mathbf{c} \in \mathbb{R}^m$ and parameter $\sigma > 0$ is defined $\rho_{\mathbf{c},\sigma}(\mathbf{x}) = \exp(-\pi||\mathbf{x} - \mathbf{c}||^2/\sigma^2)$ for $\forall$ $\mathbf{x} \in \mathbb{R}^m$, and the discrete Gaussian distribution over set $\mathcal{L}$ with the center $\mathbf{c} \in \mathbb{R}^m$ and parameter $\sigma > 0$ is defined $\mathcal{D}_{\mathcal{L},\mathbf{c},\sigma}(\mathbf{x}) = \frac{\rho_{\mathbf{c},\sigma}(\mathbf{x})}{\rho_{\mathbf{c},\sigma}(\mathcal{L})}$ for $\forall$ $\mathbf{x} \in \mathcal{L}$, where $\rho_{\mathbf{c},\sigma}(\mathcal{L}) = \sum_{\mathbf{x}\in\mathcal{L}} \rho_{\mathbf{c},\sigma}(\mathbf{x})$. Throughout the paper, we often omit $\mathbf{c}$ and write $\mathcal{D}_{\mathcal{L},\sigma}$ when $\mathbf{c} = \mathbf{0}$.

Gadget Matrix. For integers $q \geq 2$ and $n \geq 1$, Micciancio and Peikert [44] defined a special matrix (known as gadget matrix) as $\mathbf{G} := \mathbf{I}_n \otimes \mathbf{g} \in \mathbb{Z}_q^{n\times N}$ for $N = n\lceil \log q\rceil$ and $\mathbf{g} := (1, 2, \ldots, 2^{\lceil \log q\rceil -1}) \in \mathbb{Z}_q^{\lceil \log q\rceil}$, and defined the inversion function as $\mathbf{G}^{-1} : \mathbb{Z}_q^{n\times N} \to \{0,1\}^{N\times N}$. Hence, given any matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times N}$, we have $\mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{A}) = \mathbf{A}$ where $||\mathbf{G}^{-1}(\mathbf{A})||_2 \leq m$.

Learning with Errors (LWE). Given n, $q \geq 1$, $m \geq O(n \log q)$, and a discrete Gaussian distribution $\mathcal{D}_{\mathbb{Z}^m,\alpha q}$ where $0 < \alpha < 1$, the $\mathrm{LWE}_{n,q,m,\alpha}$ problem is defined to distinguish between the following two distributions:

$$(\mathbf{A}, \mathbf{A}^T\mathbf{s} + \mathbf{e}) \quad \text{and} \quad (\mathbf{A}, \mathbf{u})$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$, $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\alpha q}$, $\mathbf{u} \leftarrow \mathbb{Z}_q^m$. The LWE problem was reduced to certain worst-case lattice problems [50, 52].

Lemma 1 ([16,50]). *For all $\epsilon > 0$, there exist $q = q(n) \leq 2^n$ and a discrete Gaussian distribution $\mathcal{D}_{\mathbb{Z},\alpha q}$ satisfying $(\sqrt{n}\alpha)^{-1} \geq 2^{n^{\epsilon}}$ and $\alpha q \geq \Omega(\sqrt{n})$ such that the $\mathrm{LWE}_{n,q,m,\alpha}$ problem is at least as hard as the quantum hardness of SIVP_β and the classical hardness of GapSVP_β, where $\beta = 2^{\Omega(n^{\epsilon})}$.*

Lemma 2 ([28]). *Assume the columns of $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ generate $\mathbb{Z}_q^n$. Let $r \geq \omega(\sqrt{\log m})$. Then the distribution of the syndrome $\mathbf{u} = \mathbf{A}\mathbf{e} \pmod q$ is statistically close to uniform over $\mathbb{Z}_q^n$, where $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z}^m,r}$.*

Lemma 3 ([5]). *Let $q > 2$ and $m > (n + 1)\log q + \omega(\log n)$. For some polynomial $k = k(n)$, given $\mathbf{U} \leftarrow \{-1,1\}^{m\times k}$, $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$ and $\mathbf{B} \leftarrow \mathbb{Z}_q^{n\times k}$, we have that $(\mathbf{A}, \mathbf{A}\mathbf{U}, \mathbf{U}^T\mathbf{r}) \overset{stat}{\approx} (\mathbf{A}, \mathbf{B}, \mathbf{U}^T\mathbf{r})$ for all vectors $\mathbf{r} \in \mathbb{Z}_q^m$, where $\overset{stat}{\approx}$ indicates that the distributions are statistically indistinguishable.*

Lemma 4 ([12,28,45]). *Let n, $q > 2$, and $m > n$. Given $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a short basis $\mathbf{T}_\mathbf{A}$ for $\Lambda_q^{\perp}(\mathbf{A})$, for any $\mathbf{u} \in \mathbb{Z}_q^n$, $\mathbf{D} \in \mathbb{Z}_q^{n\times m}$ and $\sigma \geq ||\mathbf{T}_\mathbf{A}||_{\mathrm{GS}} \cdot \omega(\sqrt{\log m})$, we have*

1. $\Pr[\ \mathbf{x} \leftarrow \mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{A}),\sigma} \quad | \quad ||\mathbf{x}|| > \sqrt{m} \cdot \sigma\ ] = \mathrm{negl}(n)$.
2. $\Pr[\ \mathbf{R} \leftarrow \mathcal{D}_{\Lambda_q^{\mathbf{D}}(\mathbf{A}),\sigma} \quad | \quad ||\mathbf{R}||_2 > m \cdot \sigma\ ] = \mathrm{negl}(n)$.
3. $\Pr[\ \mathbf{S} \leftarrow \{-1,1\}^{m\times m} \quad | \quad ||\mathbf{S}||_2 > 20\sqrt{m}\ ] = \mathrm{negl}(n)$.

The following lemmas show the properties of lattice trapdoors.

Lemma 5 ([5,17,44]). *Let $n \geq 1$, $q \geq 2$, and $m = \Theta(n \log q)$. There exist several PPT algorithms with the following properties:*

- *There exists* ***TrapGen***$(1^n, 1^m, q)$ *algorithm that outputs a matrix* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ *and a short basis* $\mathbf{T_A} \in \mathbb{Z}^{m\times m}$ *for* $\Lambda_q^{\perp}(\mathbf{A})$ *such that* $\mathbf{A} \overset{stat}{\approx} \mathbf{U}$*, where* $\mathbf{U} \leftarrow \mathbb{Z}_q^{n\times m}$ *and* $||\mathbf{T_A}||_{\text{GS}} \leq O(\sqrt{n \log q})$.
- *There exists* ***SampleLeft***$(\mathbf{A}, \mathbf{B}, \mathbf{T_A}, \mathbf{D}, s)$ *algorithm that, given matrices* $\mathbf{A}, \mathbf{D} \in \mathbb{Z}_q^{n\times m}$*,* $\mathbf{B} \in \mathbb{Z}_q^{n\times m_1}$*, a short basis* $\mathbf{T_A}$ *for* $\Lambda_q^{\perp}(\mathbf{A})$ *and* $s \geq ||\mathbf{T_A}||_{\text{GS}} \cdot \omega(\sqrt{\log(m+m_1)})$*, outputs a matrix* $\mathbf{R} \in \mathbb{Z}_q^{(m+m_1)\times m}$ *from a distribution that is distributed statistically close to* $\mathcal{D}_{\Lambda_q^{\mathbf{D}}(\mathbf{A}|\mathbf{B}),s}$.
- *There exists* ***SampleRight***$(\mathbf{A}, \mathbf{G}, \mathbf{R}, \mathbf{T_G}, \mathbf{D}, \tau)$ *algorithm that, on input matrices* $\mathbf{A}, \mathbf{G} \in \mathbb{Z}_q^{n\times m}$*, a low-norm matrix* $\mathbf{R} \in \mathbb{Z}^{m\times m}$*, a basis* $\mathbf{T_G}$ *for* $\Lambda_q^{\perp}(\mathbf{G})$*, a matrix* $\mathbf{D} \leftarrow \mathbb{Z}_q^{n\times k}$ *and a parameter* $\tau \geq \sqrt{5} \cdot (||\mathbf{R}||_2 + 1) \cdot \omega(\sqrt{\log m})$*, outputs a matrix* $\mathbf{E} \in \mathbb{Z}_q^{2m\times k}$ *from a distribution that is distributed statistically close to* $\mathcal{D}_{\Lambda_q^{\mathbf{D}}(\mathbf{F}),\tau}$*, where* $\mathbf{F} := (\mathbf{A}|\mathbf{AR} + \mathbf{G})$*, where* $\mathbf{G}$ *is a gadget matrix.*
- *The lattice* $\Lambda_q^{\perp}(\mathbf{G})$ *for the gadget matrix* $\mathbf{G} \in \mathbb{Z}_q^{n\times m}$ *has a publicly known basis* $\mathbf{T_G}$ *and* $||\mathbf{T_G}||_{\text{GS}} \leq \sqrt{5}$.

Lemma 6 ([40]). *Let* $n \geq 1$*,* $q \geq 2$*,* $m = \Theta(n \log q)$*. For any* $\mathbf{R} \in \mathbb{Z}^{m\times m}$*,* $s \geq \omega(\sqrt{\log m})$*, and* $\gamma \geq s\sqrt{m}||\mathbf{R}|| \cdot \lambda^{\omega(1)}$*, the following output distributions* $(\mathbf{A}, \mathbf{AR}, \mathbf{y}, \mathbf{u})$ *are statistically close.*

- ***Sampler-1***$(\mathbf{R}, \gamma, s)$*: Given a matrix* $\mathbf{R} \in \mathbb{Z}^{m\times m}$ *and* $s, \gamma \in \mathbb{R}$*, the sampler performs the following steps:*
  1. *Run* $(\mathbf{A}, \mathbf{T_A}) \leftarrow \mathbf{TrapGen}(1^n, m, q)$.
  2. *Sample two random vectors* $\mathbf{u} \leftarrow \mathbb{Z}_q^n$ *and* $\mathbf{x} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\gamma}$.
  3. *Compute* $\mathbf{z} = \mathbf{u} - \mathbf{Ax}$.
  4. *Run* $\begin{bmatrix}\mathbf{z}_1\\ \mathbf{z}_2\end{bmatrix} \in \mathbb{Z}^{2m} \leftarrow$ ***SampleLeft***$(\mathbf{A}, \mathbf{AR}, \mathbf{T_A}, \mathbf{u}, s)$ *to generate a low-norm vector* $\begin{bmatrix}\mathbf{z}_1\\ \mathbf{z}_2\end{bmatrix}$ *such that* $(\mathbf{A}|\mathbf{AR})\begin{bmatrix}\mathbf{z}_1\\ \mathbf{z}_2\end{bmatrix} = \mathbf{z}$*. Let* $\mathbf{y} = \begin{bmatrix}\mathbf{z}_1 + \mathbf{x}\\ \mathbf{z}_2\end{bmatrix}$*, satisfying* $(\mathbf{A}|\mathbf{AR})\mathbf{y} = \mathbf{u}$.
  5. *Output the tuple* $(\mathbf{A}, \mathbf{AR}, \mathbf{y}, \mathbf{u})$.
- ***Sampler-2***$(\mathbf{R}, \gamma, s)$*: Given a matrix* $\mathbf{R} \in \mathbb{Z}^{m\times m}$ *and* $s, \gamma \in \mathbb{R}$*, the sampler performs the following steps:*
  1. *Sample a random matrix* $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$.
  2. *Sample two random vectors* $\mathbf{x} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\sqrt{\gamma^2+s^2}}$ *and* $\mathbf{z}_2 \leftarrow \mathcal{D}_{\mathbb{Z}^m,s}$.
  3. *Compute* $\mathbf{u} = \mathbf{Ax}$.
  4. *Let* $\mathbf{y} = \begin{bmatrix}\mathbf{x} - \mathbf{Rz}_2\\ \mathbf{z}_2\end{bmatrix}$*, satisfying* $(\mathbf{A}|\mathbf{AR})\mathbf{y} = \mathbf{u}$.
  5. *Output the tuple* $(\mathbf{A}, \mathbf{AR}, \mathbf{y}, \mathbf{u})$.

Lemma 7 ([12, 29]). *Let* λ *be a security parameter, and let* $(\lambda, n, m, q, \alpha)$*, where* $\alpha q \geq \Omega(\sqrt{n})$*. For any* $\mathbf{B}_1, \ldots, \mathbf{B}_\theta \leftarrow \mathbb{Z}_q^{n\times m}$*, any Boolean circuit* $f : \{0,1\}^\theta \to \{0,1\}$ *with depth* $\leq d$*, and any string* $x \in \{0,1\}^\theta$*, if*

$$\mathbf{c}_i = (\mathbf{B}_i + x_i\mathbf{G})^T\mathbf{s} + \mathbf{e}_i \ \forall i \in [\theta]$$

for $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ *and* $\mathbf{e}_i \leftarrow \mathcal{D}_{\mathbb{Z}^m,\alpha q}$ *for* $i \in [\theta]$*, then there exist (*$\mathbf{Eval}_{pk}$*,* $\mathbf{Eval}_{ct}$*,* $\mathbf{Eval}_{sim}$*) algorithms such that,*

- $\mathbf{Eval}_{pk}\big(f, (\mathbf{B}_1, \ldots, \mathbf{B}_\theta)\big) \to \mathbf{B}_f$*: Take as input* f *and* $(\mathbf{B}_1, \ldots, \mathbf{B}_\theta)$*, output a matrix* $\mathbf{B}_f \in \mathbb{Z}_q^{n\times m}$.

- $\mathbf{Eval}_{ct}\big(f, \{(\mathbf{B}_i, x_i, \mathbf{c}_i)\}_{i\in[\theta]}\big) \to \mathbf{c}_f$: *Take as input* f, $(\mathbf{B}_1, \dots, \mathbf{B}_\theta)$, *the string* x, *and* $(\mathbf{c}_1, \dots, \mathbf{c}_\theta)$, *output a vector* $\mathbf{c}_f \in \mathbb{Z}_q^m$ *that satisfies*

$$\mathbf{c}_f = (\mathbf{B}_f + f(x)\mathbf{G})^T\mathbf{s} + \mathbf{e}_f$$

  where $\mathbf{B}_f = \mathbf{Eval}_{pk}\big(f, (\mathbf{B}_1, \dots, \mathbf{B}_\theta)\big)$ *and* $||\mathbf{e}_f|| \leq \alpha q \cdot \sqrt{m} \cdot (m+1)^d$ *with all but negligible probability in* m.

- $\mathbf{Eval}_{sim}\big(f, \{(\mathbf{S}_i^*, x_i^*)\}_{i\in[\theta]}, \mathbf{A}\big) \to \mathbf{S}_f^*$: *Take as input* f, $\mathbf{S}_1^*, \dots, \mathbf{S}_\theta^* \leftarrow \mathbb{Z}_q^{m\times m}$, $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$, *and a string* $x^* \in \{0,1\}^\theta$, *output a matrix* $\mathbf{S}_f^* \in \mathbb{Z}_q^{m\times m}$ *such that*

$$\mathbf{A}\mathbf{S}_f^* - f(x^*)\mathbf{G} = \mathbf{C}_f$$

  where $\mathbf{C}_f = \mathbf{Eval}_{pk}\big(f, (\mathbf{A}\mathbf{S}_1^* - x_1^*\mathbf{G}, \dots, \mathbf{A}\mathbf{S}_\theta^* - x_\theta^*\mathbf{G})\big)$. *Moreover, if* $\mathbf{S}_1^*, \dots, \mathbf{S}_\theta^* \in \{-1,1\}^{m\times m}$, *then* $||\mathbf{S}_f^*||_2 \leq 20\sqrt{m} \cdot (m+1)^d < (m+1)^{d+1}$ *with all but negligible probability in* m.

2.2 Cryptographic Primitives

In this section, we introduce some important cryptographic primitives that we use throughout this paper.

Definition 2 (Pseudorandom Function). *Given the security parameter* λ, *a pseudorandom function (PRF) with key-space* $\mathcal{K} = \{K_\lambda\}$, *domain* $\mathcal{X} = \{X_\lambda\}$, *and range* $\mathcal{Y} = \{Y_\lambda\}$ *is an efficiently-computable function* $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ *such that for all efficient adversaries* $\mathcal{A}$, *we have that*

$$\big|\Pr[\mathcal{A}^{F(k,\cdot)}(1^\lambda) = 1] - \Pr[\mathcal{A}^{\mathcal{O}(\cdot)}(1^\lambda) = 1]\big| = \mathrm{negl}(\lambda),$$

where $k \leftarrow \mathcal{K}$ *and* $\mathcal{O}$ *is a random oracle. The probabilities are taken over all of the randomness of the experiment.*

Definition 3 (Keyed Collision-Resistant Hash Function). *Given the security parameter* λ, *a keyed collision-resistant hash function with key-space* $\mathcal{K} = \{K_\lambda\}$, *domain* $\mathcal{X} = \{X_\lambda\}$, *and range* $\mathcal{Y} = \{Y_\lambda\}$ *is an efficiently-computable function* $H : \mathcal{K}\times\mathcal{X} \to \mathcal{Y}$ *such that for all efficient adversaries* $\mathcal{A}$, *we have that*

$$\big|\Pr[(x_0, x_1) \leftarrow \mathcal{A}(1^\lambda, K) : x_0 \neq x_1 \text{ and } H(k, x_0) = H(k, x_1)]\big| = \mathrm{negl}(\lambda),$$

where $k \leftarrow \mathcal{K}$. *The probabilities are taken over all of the randomness of the experiment.*

Lemma 8 (Subset-Cover Set Systems [46]). *Let* N *be a positive integer. There exists a subset-cover set system* $[W]$ *for* $[N]$ *where* $W = 2N-1$, *with a pair of algorithms* (***Encode***, ***ComputeCover***) *satisfying the following properties:*

- ***Encode***$(x) \to \mathcal{I}_x$: *On input an element* $x \in [N]$, *the encoding algorithm outputs a set of indices* $\mathcal{I}_x \subseteq [W]$ *such that* $|\mathcal{I}_x| = \log N + 1$.
- ***ComputeCover***$(\mathcal{L}) \to \mathcal{J}_\mathcal{L}$: *On input a revocation list* $\mathcal{L} \subseteq [N]$, *the cover computation algorithm outputs a collection of indices* $\mathcal{J}_\mathcal{L} \subseteq [W]$ *such that* $|\mathcal{J}_\mathcal{L}| = O(|\mathcal{L}| \log(N/|\mathcal{L})|)$.
- ***Correctness.*** *For any element* $x \in [N]$ *and revocation list* $\mathcal{L} \subseteq [N]$, *it holds that* $x \in \mathcal{L}$ *iff.* $\mathcal{I}_x \cap \mathcal{J}_\mathcal{L} = \emptyset$, *where* $\mathcal{I}_x \leftarrow$ ***Encode***(x) *and* $\mathcal{J}_\mathcal{L} \leftarrow$ ***ComputeCover***$(\mathcal{L})$.

2.3 Cryptographic Problems

Definition 4 (Generalized Jump-Finding Problem [48]). *For positive integers $N, r, q \in \mathbb{N}$ and $\delta, \varepsilon > 0$, the $(N, r, q, \delta, \varepsilon)$-generalized jump-finding problem is defined as follows. An adversary begins by choosing a set C of up to q tuples $(s, b_1, \ldots, b_r) \in [N] \times \{0,1\}^r$ where all of the s are distinct. Each tuple $(s, b_1, \ldots, b_r)$ describes a curve between grid points from the top to bottom of the grid $[1, r] \times [0, 2N]$, which oscillates about the column at position $2s - 1$, with $b = (b_1, \ldots, b_r)$ specifying which side of the column the curve is on in each row. The curves divide the grid into $|C| + 1$ contiguous regions. For each pair $(i, x) \in [1, r] \times [0, 2N]$, the adversary chooses a probability $p_{i,x} \in [0, 1]_{\mathbb{R}}$ with the following properties:*

- *For any two pairs $(i, 2x), (j, 2x) \in [1, r] \times [0, 2N]$, it holds that $|p_{i,2x} - p_{j,2x}| \leq \delta$.*
- *Let $C_i = \{(s, b_1, \ldots, b_r) \in C : 2s - b_i\}$ be the set of values $2s - b_i$ for tuples in C. For any two pairs $(i, x), (i, y) \in [1, r] \times [0, 2N]$ such that $(x, y] \cap C_i = \emptyset$, then $|p_{i,x} - p_{i,y}| < \delta$.*
- *For all $i, j \in [r]$, it holds that $p_{i,0} = p_{j,0}$ and $p_{i,2N} = p_{j,2N}$. Define $p_0 = p_{i,0}$ and $p_{2N} = p_{i,2N}$.*
- *Finally, $|p_{2N} - p_0| > \varepsilon$.*

Note that the above properties indicate that $p_{i,x}$ varies "minimally" within each contiguous region but overall from left to right, there is "significant" variation of the $p_{i,x}$. Next, define the oracle $\mathcal{Q} : [1, r] \times [0, 2N] \to \{0, 1\}$ to be a randomized oracle that on input (i, x) outputs 1 with probability $p_{i,x}$. Repeated calls to $\mathcal{Q}$ on the same input (i, x) will yield a fresh and independently-sampled bit. The $(N, r, q, \delta, \varepsilon)$-generalized jump-finding problem is to output some element in C given oracle access to $\mathcal{Q}$.

Lemma 9 (Generalized Jump-Finding Problem [48]). *There is an efficient algorithm **QTrace**$^{\mathcal{Q}}$ $(\lambda, N, r, q, \delta, \varepsilon)$ that runs in time $t = \mathsf{poly}(\lambda, \log N, r, q, 1/\delta)$ and makes at most t queries to $\mathcal{Q}$ that solves the $(N, r, q, \delta, \varepsilon)$-generalized jump-finding problem with probability $1 - \mathrm{negl}(\lambda)$ whenever $\varepsilon \geq \delta(9 + 4(\lceil \log N \rceil - 1)q)$. Moreover, any element $(s, b_1, \ldots, b_r) \in [N] \times \{0,1\}^r$ output by **QTrace**$^{\mathcal{Q}}$ satisfies the following property (with overwhelming probability):*

- *For all $i \in [r]$, $|P(i, 2s - b_i) - P(i, 2s - 1 - b_i)| \geq \delta$, where $P(i, x) := \Pr[\mathcal{Q}(i, x) = 1]$.*

2.4 Functional Encryption

In this section, we recall the notion of mixed functional encryption (mixed FE) that we use in this work.

Definition 5 (Mixed FE [30]). *A mixed functional encryption scheme $\Pi_{\boldsymbol{MFE}}$ with domain $\mathcal{X}$ and function family $\mathcal{F} = \{f : \mathcal{X} \to \{0, 1\}\}$ is a tuple of algorithms $\Pi_{\boldsymbol{MFE}} =$ (**PrmsGen**, **MSKGen**, **KeyGen**,**PKEnc**, **SKEnc**, **Dec**):*

- ***PrmsGen**(1^λ): On input the security parameter λ, output the public parameters pp.*
- ***MSKGen**(pp): On input pp, output a master secret key msk.*
- ***KeyGen**(msk, x): On input msk and an input $x \in \mathcal{X}$, output a secret key sk_x.*
- ***PKEnc**(pp): On input pp, output a ciphertext ct.*
- ***SKEnc**(msk, f): On input msk and a function $f \in F$, output a ciphertext ct_f.*
- ***Dec**$(\mathsf{sk}_x, \mathsf{ct})$: On input sk_x and ct, output a bit $b \in \{0, 1\}$.*

A mixed FE scheme should satisfy the following properties:

- ***Correctness**: For all functions $f \in F$ and all inputs $x \in \mathcal{X}$, let $\mathsf{pp} \leftarrow$ **PrmsGen**(1^λ), $\mathsf{msk} \leftarrow$ **MSKGen**(pp), $\mathsf{sk}_x \leftarrow$ **KeyGen**(msk, x), $\mathsf{ct} \leftarrow$ **PKEnc**(pp), and $\mathsf{ct}_f \leftarrow$ **SKEnc**(msk, f), we have*

$$\Pr[\mathbf{Dec}(\mathsf{sk}_x, \mathsf{ct}) = 1] = 1 - \mathrm{negl}(\lambda) \text{ and } \Pr[\mathbf{Dec}(\mathsf{sk}_x, \mathsf{ct}_f) = f(x)] = 1 - \mathrm{negl}(\lambda).$$

- ***Semantic security:*** *For a bit $b \in \{0,1\}$, we define the security experiment $\mathsf{ExptMFE}_{\mathsf{SS}}[\lambda, \mathcal{A}, b]$ that describes the interaction between a challenger and a PPT adversary $\mathcal{A}$. The challenger begins by sampling $\mathsf{pp} \leftarrow \boldsymbol{PrmsGen}(1^\lambda)$, $\mathsf{msk} \leftarrow \boldsymbol{MSKGen}(\mathsf{pp})$, and gives pp to $\mathcal{A}$. The adversary $\mathcal{A}$ is then given access to the following oracles:*
  - ***Key query:*** *Whenever $\mathcal{A}$ submits $x \in \mathcal{X}$, the challenger replies with $\mathsf{sk}_x \leftarrow \boldsymbol{KeyGen}(\mathsf{msk}, x)$.*
  - ***Encryption query:*** *Whenever $\mathcal{A}$ submits $f \in F$, the challenger replies with $\mathsf{ct}_f \leftarrow \boldsymbol{SKEnc}(\mathsf{msk}, f)$.*
  - ***Challenge query:*** *$\mathcal{A}$ submits two functions $f_0, f_1 \in F$ such that $f_0(x) = f_1(x)$ for all $x \in \mathcal{X}$ the adversary $\mathcal{A}$ submitted to the key generation oracle, the challenger replies with $\mathsf{ct}_{f_b} \leftarrow \boldsymbol{SKEnc}(\mathsf{msk}, f_b)$ for $b \leftarrow \{0,1\}$. Note that the challenge query can be made only once.*

  At the end of the experiment, the adversary outputs a bit $b' \in \{0,1\}$, which is also the output of the experiment. We say $\Pi_{\boldsymbol{MFE}}$ satisfies (adaptive) semantic security if for all PPT $\mathcal{A}$, we have
  $$|\Pr[\mathsf{ExptMFE}_{\mathsf{SS}}[\lambda, \mathcal{A}, 0] = 1] - \Pr[\mathsf{ExptMFE}_{\mathsf{SS}}[\lambda, \mathcal{A}, 1] = 1]| = \mathrm{negl}(\lambda).$$
- ***Public/secret key indistinguishability:*** *For a bit $b \in \{0,1\}$, we define the security experiment $\mathsf{ExptMFE}_{\mathsf{PK/SK}}[\lambda, \mathcal{A}, b]$ that describes the interaction between a challenger and a PPT adversary $\mathcal{A}$ exactly as $\mathsf{ExptMFE}_{\mathsf{SS}}[\lambda, \mathcal{A}, b]$, except the challenge oracle is replaced with the following:*
  - ***Challenge query:*** *$\mathcal{A}$ submits $f^* \in F$ such that $f^*(x) = 1$ for all $x \in \mathcal{X}$ the adversary $\mathcal{A}$ submitted to the key generation oracle, the challenger computes $\mathsf{ct}_0 \leftarrow \boldsymbol{PKEnc}(\mathsf{pp})$, $\mathsf{ct}_1 \leftarrow \boldsymbol{SKEnc}(\mathsf{msk}, f^*)$ and returns ct_b for $b \leftarrow \{0,1\}$ to $\mathcal{A}$. Note that the challenge oracle can be made only once.*

  At the end of the game, the adversary outputs a bit $b' \in \{0,1\}$, which is also the output of the game. We say $\Pi_{\boldsymbol{MFE}}$ satisfies (adaptive) public/secret key indistinguishability if for all PPT $\mathcal{A}$, we have
  $$|\Pr[\mathsf{ExptMFE}_{\mathsf{PK/SK}}[\lambda, \mathcal{A}, 0] = 1] - \Pr[\mathsf{ExptMFE}_{\mathsf{PK/SK}}[\lambda, \mathcal{A}, 1] = 1]| = \mathrm{negl}(\lambda).$$

Definition 6 (Non-Adaptive q-Query Security). *For each of the security notions in Definition 5 (semantic security, public/secret key indistinguishability), we define a notion of non-adaptive q-query security where the corresponding security notion only holds against all PPT adversaries that make at most $q \in \mathbb{N}$ queries to the encryption oracle, and moreover, all of the non-encryption queries occur before the encryption queries.*

As stated in [39], assuming the sub-exponential hardness of LWE (with super-polynomial modulus-to-noise ratio), the construction of [30] gives a mixed FE scheme that supports the class of $\mathbf{NC}^1$ functions and satisfies non-adaptive q-query security for any a priori bounded $q = \mathbf{poly}(\lambda)$, and the construction of [18] gives a simpler mixed FE scheme that supports all circuits of a priori bounded polynomial depth $d = d(\lambda)$ and satisfies non-adaptive q-query security for a priori bounded $q = q(\lambda)$.

2.5 Attribute-Based Functional Encryption

In this section, we review the notion of attribute-based functional encryption (ABFE) and then describe a concrete construction of attribute-based inner-product functional encryption (ABIPFE) based on LWE.

Definition 7 (ABFE). *Given a message space $\mathcal{M}$, an attribute space $\mathcal{X}$, and two family of functions $\mathcal{F} = \{f : \mathcal{X} \to \{0,1\}\}$, $\mathcal{G}_\lambda = \{g : \mathcal{X}_\lambda \to \mathcal{Z}_\lambda\}$, where $\mathcal{M} \subseteq \mathcal{X}_\lambda$, an attribute-based functional encryption scheme contains a tuple of algorithms $\Pi_{\boldsymbol{ABFE}} = (\boldsymbol{Setup}, \boldsymbol{KeyGen}, \boldsymbol{Enc}, \boldsymbol{Dec})$:*

- ***Setup***(1^λ). *Take as input a security parameter* λ, *output a master public/secret key pair* $(\mathsf{mpk}, \mathsf{msk})$.
- ***KeyGen***(msk, f, g). *Take as input* msk, $f \in \mathcal{F}$, *and* $g \in \mathcal{G}_\lambda$, *output a secret key* $\mathsf{sk}_{f,g}$.
- ***Enc***$(\mathsf{mpk}, \mathsf{att}, \mu)$. *Take as input* mpk, *attribute* $\mathsf{att} \in \mathcal{X}$, *and message* $\mu \in \mathcal{M}$, *output a ciphertext* ct.
- ***Dec***$(\mathsf{sk}, \mathsf{ct})$. *Take as input* sk *and* ct, *output* $g(\mu)$ *or* $\perp$.

An attribute-based functional encryption scheme should satisfy the following properties:

- ***Correctness****: For all functions* $f \in \mathcal{F}, g \in \mathcal{G}_\lambda$, *all attributes* $\mathsf{att} \in \mathcal{X}$ *such that* $f(\mathsf{att}) = 0$, *and all messages* $\mu \in \mathcal{M}$, *let* $(\mathsf{mpk}, \mathsf{msk}) \leftarrow$ ***Setup***(1^λ), $\mathsf{sk}_{f,g} \leftarrow$ ***KeyGen***(msk, f, g), *we have*

$$\Pr[\mathbf{Dec}\big(\mathsf{sk}_{f,g}, \mathbf{Enc}(\mathsf{mpk}, \mathsf{att}, \mu)\big) = g(\mu)] = 1 - \mathrm{negl}(\lambda).$$

- ***Adaptive security****: We define the security experiment that describes the interaction between a PPT adversary* $\mathcal{A}$ *and a challenger. The challenger begins by sampling* $(\mathsf{mpk}, \mathsf{msk}) \leftarrow$ ***Setup***(1^λ) *and gives* mpk *to* $\mathcal{A}$. *The adversary* $\mathcal{A}$ *is then given access to the following oracles:*
  - ***Key query*** $\mathcal{O}_{\boldsymbol{KG}}$*: Whenever* $\mathcal{A}$ *submits a pair* $(f, g) \in \mathcal{F} \times \mathcal{G}_\lambda$, *the challenger responds with* $\mathsf{sk}_{f,g} \leftarrow$ ***KeyGen***(msk, f, g).
  - ***Challenge query****:* $\mathcal{A}$ *submits two messages* (μ_0, μ_1) *and a challenge attribute* att^* *such that for all pairs* (f, g) *the adversary* $\mathcal{A}$ *submitted to* $\mathcal{O}_{KG}$, *it holds that* $f(\mathsf{att}^*) = 1$ *or* $(f(\mathsf{att}^*) = 0, g(\mu_0) = g(\mu_1))$. *The challenger replies with* $\mathsf{ct}_b \leftarrow \mathbf{Enc}(\mathsf{mpk}, \mathsf{att}^*, \mu_b)$ *for* $b \leftarrow \{0, 1\}$. *Note that the challenge query can be made only once.*

  At the end of the game, the adversary outputs a bit $b' \in \{0, 1\}$, *which is also the output of the experiment. The advantage of* $\mathcal{A}$ *in winning the above security game is defined as:* $\mathrm{Adv}^{\mathrm{AS}}_{\mathrm{ABFE},\mathcal{A}}(1^\lambda) := |\Pr[b' = b] - 1/2|$. *We say* $\Pi_{\boldsymbol{ABFE}}$ *satisfies adaptive security if for all PPT* $\mathcal{A}$, *we have* $\mathrm{Adv}^{\mathrm{AS}}_{\mathrm{ABFE},\mathcal{A}}(1^\lambda) = \mathrm{negl}(\lambda)$.

Weaker than the security model defined above are 1) semi-adaptive security model, where $\mathcal{A}$ announces the challenge attribute att^* on receipt of the master public key but before launching any key query, and 2) selective security model, where $\mathcal{A}$ announces the challenge attribute att^* at the beginning of the experiment.

Definition 8 (ABIPFE). *An ABIPFE is simply an ABFE scheme where we let* $\mathcal{G}_\lambda$ *be a family of inner-product functions. To be precise, its syntax and selective/semi-adaptive/adaptive security correspond to the corresponding definitions above, except with some substitutions:* 1) *let* $\mathcal{G}_\lambda = \{\mathsf{IP} : \mathcal{X}_\lambda \times \mathcal{Y}_\lambda \to \mathcal{Z}_\lambda\}$, *and* 2) *replace all* g *with* $y \in \mathcal{Y}_\lambda$, *all* μ *with* $x \in \mathcal{X}_\lambda$, *and all* $g(\mu)$ *with* $\mathsf{IP}(x, y)$.

Remark 1 (Comparison to Previous Security Notions). Our selective security definition combines the AD-CPA security of IPFE (see Appendix A) with the selective security definition of ABE [12]. We note that three previous constructions on ABIPFE consider different selective security definitions that are weaker than ours. The first one was considered by Luo and Al-Kuwari [42], which they called weakly selective security due to the following constraints: 1) every f of the pair (f, y) of $\mathcal{O}_{\mathrm{KG}}$ can only be queried once, and 2) all pairs (f, y) of $\mathcal{O}_{\mathrm{KG}}$ must satisfy $f(\mathsf{att}^*) = 1$. Pal and Dutta [49] defined a different (weak) selective security, where $\mathcal{A}$ must announce an attribute-function pair (att^*, f^*) (or multiple functions) such that $f^*(\mathsf{att}^*) = 0$ beforehand, and all pairs (f, y) of $\mathcal{O}_{\mathrm{KG}}$ must satisfy one of the following conditions: 1) $f(\mathsf{att}^*) = 1$; 2) $\mathsf{IP}(x_0, y) = \mathsf{IP}(x_1, y)$ and $f = f^*$. Compared to the above two (weak) selective security, the selective security notion considered by Lai *et al.* [40] is stronger, but still somewhat weaker than ours. Specifically, their selective security definition must be subject to the following restrictions (vs. ours): 1) $\mathcal{A}$ is allowed to make no more than $Q \in \mathbb{N}$ key queries for the pair $(f, y) \in \mathcal{F} \times \mathcal{G}_\lambda$ that satisfies $(f(\mathsf{att}^*) = 0, \mathsf{IP}(x_0, y) = \mathsf{IP}(x_1, y))$, and 2) $\mathcal{A}$ is stateful, that is, it carries states during all stages of the interaction with the challenger.

3 Attribute-Based Inner-Product Functional Encryption based on LWE

Our ABIPFE: Given a family of functions $\mathcal{F} = \{f : \{0,1\}^\theta \to \{0,1\}\}$ of depth $\leq d$, an attribute space $\mathcal{X} = \{0,1\}^\theta$, a message space $\mathcal{X}_\lambda = \{0,\ldots,P-1\}^m$, a key space $\mathcal{Y}_\lambda = \{0,\ldots,V-1\}^m$, and an output space $\mathcal{Z}_\gamma = \{0,\ldots,K-1\}$, where $K = mPV$, our ABIPFE construction works for any $\theta, d = \text{poly}(\lambda)$:

- **Setup**$(1^\lambda, 1^\theta)$. Run **TrapGen**$(1^n, m, q) \to (\mathbf{A}, \mathbf{T_A})$. Sample $(\theta+1)$ matrices $\mathbf{U}, \mathbf{B}_1, \ldots, \mathbf{B}_\theta \leftarrow \mathbb{Z}_q^{n\times m}$. Output $mpk := (\mathbf{A}, \mathbf{U}, \mathbf{B}_1, \ldots, \mathbf{B}_\theta, \mathbf{G})$ and $msk := (\mathbf{T_A})$.
- **KeyGen**$(msk, f \in \mathcal{F}, \mathbf{y} \in \mathcal{Y}_\lambda)$. Let $\mathbf{B}_f = \mathbf{Eval}_{pk}(f, (\mathbf{B}_1, \ldots, \mathbf{B}_\theta))$. Choose $\mathbf{W} \leftarrow \mathcal{D}_{\mathbb{Z}^{m\times m},\gamma}$ and let $\mathbf{D} = \mathbf{U} - \mathbf{AW}$. Run $\begin{bmatrix}\mathbf{R}_1\\ \mathbf{R}_2\end{bmatrix} \in \mathbb{Z}^{2m\times m} \leftarrow$ **SampleLeft**$(\mathbf{A}, \mathbf{B}_f, \mathbf{T_A}, \mathbf{D}, s)$ to produce a low-norm matrix $\begin{bmatrix}\mathbf{R}_1\\ \mathbf{R}_2\end{bmatrix}$ such that $(\mathbf{A}|\mathbf{B}_f)\begin{bmatrix}\mathbf{R}_1\\ \mathbf{R}_2\end{bmatrix} = \mathbf{D}$. Let $\mathbf{R}_f = \begin{bmatrix}\mathbf{R}_1 + \mathbf{W}\\ \mathbf{R}_2\end{bmatrix}$. Output a secret key $sk_{f,\mathbf{y}} := (\mathbf{y}, \mathbf{R}_f \cdot \mathbf{y})$.
- **Enc**$(mpk, \mathbf{x} \in \mathcal{X}_\lambda, \mathsf{att} \in \{0,1\}^\theta)$. Choose $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $\mathbf{e}_0, \mathbf{e}_1 \leftarrow \mathcal{D}_{\mathbb{Z}^m,\sigma}$, and θ matrices $\mathbf{S}_i \leftarrow \{\pm 1\}^{m\times m}$. Set

$$\mathbf{H} = (\mathbf{A} \mid \mathbf{B}_1 + \mathsf{att}_1 \cdot \mathbf{G} \mid \cdots \mid \mathbf{B}_\theta + \mathsf{att}_\theta \cdot \mathbf{G}),$$
$$\mathbf{e} = (\mathbf{I}_m \mid \mathbf{S}_1 \mid \cdots \mid \mathbf{S}_\theta)^T \cdot \mathbf{e}_0.$$

  Output a ciphertext $\mathbf{c} = (\mathbf{H}^T\mathbf{s} + \mathbf{e}, \mathbf{U}^T\mathbf{s} + \mathbf{e}_1 + \lfloor q/K \rfloor \cdot \mathbf{x}) \in \mathbb{Z}_q^{(\theta+2)m}$.
- **Dec**$(sk_{f,\mathbf{y}}, \mathbf{c}, \mathsf{att} \in \{0,1\}^\theta)$. Let $\mathbf{c} = (\mathbf{c}_{in}, \mathbf{c}_1, \ldots, \mathbf{c}_\theta, \mathbf{c}_{out}) \in \mathbb{Z}_q^{(\theta+2)m}$. Compute

$$\mathbf{c}_f = \mathbf{Eval}_{ct}(f, \{(\mathbf{B}_i, \mathsf{att}_i, \mathbf{c}_i)\}_{i\in[\theta]}) \in \mathbb{Z}_q^m.$$

  Let $\mathbf{c}'_f = (\mathbf{c}_{in}|\mathbf{c}_f) \in \mathbb{Z}_q^{2m}$ and compute $\mu' = \mathbf{y}^T\mathbf{c}_{out} - (\mathbf{R}_f \cdot \mathbf{y})^T\mathbf{c}'_f \pmod q$. Output the value $\mu \in \{-K+1, \ldots, K-1\}$ that minimizes $|\lfloor q/K \rfloor \cdot \mu - \mu'|$.

Choice of Parameters: To satisfy the following correctness and security requirements, and also based on the choice of parameters of IPFE scheme of Agrawal *et al.* [6] (ALS for short) reviewed in Appendix A, we set $\lambda = n$, $\sigma = \omega(\sqrt{\log n})$, $m = 2n\log q$, $s = O\big((m+1)^{d+3/2}\big)$, $\gamma = s\sqrt{m} \cdot O\big((m+1)^{d+3/2}\big) \cdot \lambda^{\omega(1)}$, and $q = KV\omega((m+1)^{4d+7} \cdot \sqrt{\log n}) \cdot \lambda^{\omega(1)}$.

Correctness: Due to the correctness of $\mathbf{Eval}_{ct}$ algorithm of Lemma 7, we have $\mathbf{c}_f = \mathbf{B}_f^T\mathbf{s} + \mathbf{e}_f$ when $f(\mathsf{att}) = 0$, where $||\mathbf{e}_f|| \leq 20\sigma m \cdot (m+1)^d$. Consequently,

$$\mathbf{c}'_f = (\mathbf{c}_{in}|\mathbf{c}_f) = (\mathbf{A}|\mathbf{B}_f)^T\mathbf{s} + \mathbf{e}'_f,$$

where $||\mathbf{e}'_f|| < 20\sigma \cdot (m+1)^{d+1}$. Recall that $(\mathbf{A}|\mathbf{B}_f) \cdot \mathbf{R}_f = (\mathbf{A}|\mathbf{B}_f) \cdot \begin{bmatrix}\mathbf{R}_1 + \mathbf{W}\\ \mathbf{R}_2\end{bmatrix} = \mathbf{D} + \mathbf{AW} = \mathbf{U}$, where $||\mathbf{R}_f|| \leq \sqrt{m(2s^2 + \gamma^2)}$ with overwhelming probability. Therefore, it holds that

$$\begin{aligned}\mathbf{y}^T\mathbf{c}_{out} - (\mathbf{R}_f \cdot \mathbf{y})^T\mathbf{c}'_f &= (\mathbf{y}^T\mathbf{D}^T\mathbf{s} + \mathbf{y}^T\mathbf{e}_1 + \lfloor q/K \rfloor \cdot \mathbf{y}^T\mathbf{x}) - (\mathbf{y}^T\mathbf{D}^T\mathbf{s} + (\mathbf{R}_f\mathbf{y})^T\mathbf{e}'_f)\\ &= \lfloor q/K \rfloor \cdot \mathbf{y}^T\mathbf{x} + \mathbf{y}^T\mathbf{e}_1 - (\mathbf{R}_f\mathbf{y})^T\mathbf{e}'_f,\end{aligned}$$

where $|\mathbf{y}^T\mathbf{e}_1 - (\mathbf{R}_f\mathbf{y})^T\mathbf{e}'_f| < mV\sigma + 20\sqrt{2s^2 + \gamma^2}m\sigma sV(m+1)^{d+1} \leq \lfloor q/K \rfloor/4$ with overwhelming probability, which thereby ensures correct decryption of $\mathbf{y}^T\mathbf{x} \in \mathcal{Z}_\lambda$. Thus, for any $d = \text{poly}(\lambda)$, we have $2^{n^\epsilon} > KV\omega((m+1)^{4d+7})$ by setting $n = \tilde{O}(d)^{1/\epsilon}$ for $0 < \epsilon < 1$, and hence we need to rely on sub-exponential LWE with $(\alpha = 1/(\sqrt{n} \cdot 2^{n^\epsilon}), q = 2^{n^\epsilon} \cdot \Omega(n))$, which is at least as hard as SIVP_β and GapSVP_β for $\beta = 2^{\Omega(n^\epsilon)}$ by Lemma 1.

Before discussing how to construct an ABIPFE that satisfies (semi-) adaptive security, we first formally prove that the above ABIPFE scheme is selectively secure as per our selective security definition.

Theorem 1 (Selective Security). *Given* $(\mathbf{Eval}_{pk}, \mathbf{Eval}_{ct}, \mathbf{Eval}_{sim})$ *algorithms (see Lemma 5), for any family of functions* $\mathcal{F}$*, the ABIPFE scheme above is selectively secure with respect to* $\mathcal{F}$*, assuming the hardness of the* $\mathrm{LWE}_{n,m,q,\alpha}$ *problem.*

Due to space constrains, we defer the full security proof for Theorem 1 to Appendix B.

(Semi-) Adaptively Secure ABIPFE. To enhance the security of our LWE-based ABIPFE scheme, let's focus on the security proof. Since the difference between these security notions are the time at which the adversary announces the challenge attribute att^*, the method of embedding att^* into the public parameters, and the restricted conditions related to att^* that the key query must satisfy, we only need to deal with issues (e.g. public parameters, key query, and challenge ciphertext) related to att^*.

From Theorem 1, for any pair $(f, \mathbf{y})$ of $\mathcal{O}_{\mathrm{KG}}$, it suffices to consider the case that $f(\mathsf{att}^*) = 1$, as the other case only relies on the two-stage sampling technique of Lemma 6. On the other hand, it is not hard to see that the framework of our ABIPFE and its security proof for the case of $f(\mathsf{att}^*) = 1$ follow that of the LWE-based ABE scheme of [12], except that we use the two-stage sampling algorithm. Hence, any techniques that improve [12] are also applicable to our ABIPFE. We note that Brakerski and VinodVaikuntanathan [16] improved the work of [12] and proposed a semi-adaptively secure ABE scheme based on LWE by "programming" the challenge attribute att^* into the PRF values. Therefore, we can convert our selectively secure ABIPFE to the one that satisfies semi-adaptive security using the technique proposed in [16].

Moreover, like [12], we can easily obtain an adaptively secure ABIPFE from our ABIPFE by a standard complexity leveraging argument. We remark that an adaptively secure LWE-based ABIPFE without relying on the complexity leveraging technique implies an adaptively secure LWE-based ABE without relying on the complexity leveraging technique. The latter is, to the best of our knowledge, a long-standing open problem in LWE-based ABE system.

Comparison with Previous Work. Structurally, our scheme is similar to the ABIPFE schemes proposed by Luo and Al-Kuwari [42], Pal and Dutta [49], and Lai *et al.* [40]. This should not be surprising since those solutions and ours are built upon the LWE-based ABE scheme of [12] and ALS [6]. However, those of [40,42,49] only satisfy different weaker versions of the selective security (see Remark 1); by contrast, our scheme is proven selectively secure. By applying the transformation approach discussed above to [40,42,49], we can get several different ABIPFE schemes that satisfy the weaker versions of the (semi-) adaptive security accordingly. We note that Lai *et al.* [40] also presented a similar approach to upgrade ABIPFE scheme from its weaker version of the selective security to the semi-adaptive security (which is a weaker version as against ours). Looking ahead, the instantiation of our secret-key revocable predicate inner-product functional encryption with broadcast (and hence our trace-and-revoke IPFE) from LWE, which we will cover in the next section, requires an adaptively secure ABIPFE. This means that we cannot directly instantiate our trace-and-revoke IPFE using those schemes in [40,42,49].

4 Revocable Predicate Functional Encryption with Broadcast

In this section, we will show how to construct a secret-key revocable predicate functional encryption (RPFE) with broadcast in a generic way. As discussed in Section 1, this construction can be seen as a generalization of the construction of revocable predicate encryption with broadcast of [39], and the challenge is in how to instantiate it from standard assumptions.

4.1 Definition of Revocable Predicate Functional Encryption with Broadcast

Definition 9 (Secret-Key RPFE with Broadcast). *A secret-key revocable predicate functional encryption with broadcast for an identity space $\mathcal{ID}$, an attribute space $\mathcal{X}$, a message space $\mathcal{M}$, two function families $\mathcal{F} = \{f : \mathcal{X} \to \{0,1\}\}$ and $\mathcal{G}_\lambda = \{g : \mathcal{X}_\lambda \to \mathcal{Z}_\lambda\}$ ($\mathcal{M} \subseteq \mathcal{X}_\lambda$) contains a tuple of algorithms $\Pi_{\boldsymbol{RPFE}} = (\boldsymbol{Setup}, \boldsymbol{KeyGen}, \boldsymbol{Broadcast}, \boldsymbol{Enc}, \boldsymbol{Dec})$:*

- $\boldsymbol{Setup}(1^\lambda)$*. On input a security parameter λ, output the public parameters* pp *and the master secret key* msk*.*
- $\boldsymbol{KeyGen}(\mathsf{msk}, \mathsf{id}, g, x)$*. On input* msk*, an identity* $\mathsf{id} \in \mathcal{ID}$*, a function* $g \in \mathcal{G}_\lambda$*, and an attribute* $x \in \mathcal{X}$*, output a secret key* $\mathsf{sk}_{\mathsf{id},g,x}$*.*
- $\boldsymbol{Broadcast}(\mathsf{pp}, \mu, \mathcal{R})$*: On input* pp*, a message* $\mu \in \mathcal{M}$*, and a revocation list* $\mathcal{R} \subseteq \mathcal{ID}$*, output a ciphertext* $\mathsf{ct}_{\mu,\mathcal{R}}$*.*
- $\boldsymbol{Enc}(\mathsf{pp}, \mathsf{msk}, \mu, \mathcal{R}, f)$*. On input* pp*,* msk*, a message* $\mu \in \mathcal{M}$*, a revocation list* $\mathcal{R} \subseteq \mathcal{ID}$*, and a function* $f \in \mathcal{F}$*, output a ciphertext* $\mathsf{ct}_{\mu,\mathcal{R},f}$*.*
- $\boldsymbol{Dec}(\mathsf{sk}, \mathsf{ct})$*. On input* sk *and* ct*, output* $g(\mu)$ *or* $\perp$*.*

A secret-key revocable predicate functional encryption with broadcast should satisfy the following properties:

- ***Correctness****: For all identities* $\mathsf{id} \in \mathcal{ID}$*, all functions* $f \in \mathcal{F}, g \in \mathcal{G}_\lambda$*, all attributes* $x \in \mathcal{X}$ *such that* $f(x) = 1$*, all messages* $\mu \in \mathcal{M}$*, and all revocation lists* $\mathcal{R} \subseteq \mathcal{ID}$ *such that* $\mathsf{id} \notin \mathcal{R}$*, let* $(\mathsf{pp}, \mathsf{msk}) \leftarrow \boldsymbol{Setup}(1^\lambda)$ *and* $\mathsf{sk}_{\mathsf{id},g,x} \leftarrow \boldsymbol{KeyGen}(\mathsf{msk}, \mathsf{id}, g, x)$*, the following holds:*
  - ***Broadcast correctness****: For* $\mathsf{ct}_{\mu,\mathcal{R}} \leftarrow \boldsymbol{Broadcast}(\mathsf{pp}, \mu, \mathcal{R})$*, we have*

$$\Pr[\mathbf{Dec}\big(\mathsf{sk}_{\mathsf{id},g,x}, \mathsf{ct}_{\mu,\mathcal{R}}\big) = g(\mu)] = 1 - \mathrm{negl}(\lambda).$$

  - ***Encryption correctness****:For* $\mathsf{ct}_{\mu,\mathcal{R},f} \leftarrow \boldsymbol{Enc}(\mathsf{pp}, \mathsf{msk}, \mu, \mathcal{R}, f)$*, we have*

$$\Pr[\mathbf{Dec}\big(\mathsf{sk}_{\mathsf{id},g,x}, \mathsf{ct}_{\mu,\mathcal{R}}\big) = g(\mu)] = 1 - \mathrm{negl}(\lambda).$$

- ***Adaptive security****: For a bit* $b \in \{0,1\}$*, we define the security experiment* $\mathsf{ExptPRFE}_{\mathsf{AS}}[\lambda, \mathcal{A}, b]$ *that describes the interaction between a challenger and a PPT adversary* $\mathcal{A}$*. The challenger begins by sampling* $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \boldsymbol{Setup}(1^\lambda)$ *and gives* mpk *to* $\mathcal{A}$*. The adversary* $\mathcal{A}$ *is then given access to the following oracles:*
  - ***Key query*** $\mathcal{O}_{\boldsymbol{KG}}$*: Whenever* $\mathcal{A}$ *submits a triple* $(\mathsf{id}, g, x) \in \mathcal{ID} \times \mathcal{G}_\lambda \times \mathcal{X}$*, the challenger responds with* $\mathsf{sk}_{\mathsf{id},g,x} \leftarrow \boldsymbol{KeyGen}(\mathsf{msk}, \mathsf{id}, g, x)$*.*
  - ***Encryption oracle****: Whenever* $\mathcal{A}$ *submits a triple* $(\mu, \mathcal{R}, f) \in \mathcal{M} \times \mathcal{ID} \times \mathcal{F}$*, the challenger replies with* $\mathsf{ct}_{\mu,\mathcal{R},f} \leftarrow \boldsymbol{Enc}(\mathsf{pp}, \mathsf{msk}, \mu, \mathcal{R}, f)$*.*
  - ***Challenge query****:* $\mathcal{A}$ *submits two messages* (μ_0, μ_1)*, a revocation list* $\mathcal{R}^* \subseteq \mathcal{ID}$*, and a function* $f^* \in \mathcal{F}$ *such that for all triples* (id, g, x) *the adversary* $\mathcal{A}$ *submitted to* $\mathcal{O}_{KG}$*, one of the following cases holds:* 1) $f^*(x) = 0$*, or* 2) $\mathsf{id} \in \mathcal{R}^*$*, or* 3) $(f^*(x) = 1, \mathsf{id} \notin \mathcal{R}^*, g(\mu_0) = g(\mu_1))$*. The challenger replies with* $\mathsf{ct}_b \leftarrow \boldsymbol{Enc}(\mathsf{pp}, \mathsf{msk}, \mu_b, \mathcal{R}^*, f^*)$ *for* $b \leftarrow \{0,1\}$*. Note that the challenge query can be made only once.*

At the end of the experiment, the adversary outputs a bit $b' \in \{0,1\}$, which is also the output of the experiment. We say $\Pi_{\mathbf{RPFE}}$ satisfies adaptive security if for all PPT $\mathcal{A}$, we have

$$|\Pr[\mathsf{ExptPRFE}_{\mathsf{AS}}[\lambda, \mathcal{A}, 0] = 1] - \Pr[\mathsf{ExptPRFE}_{\mathsf{AS}}[\lambda, \mathcal{A}, 1] = 1]| = \mathrm{negl}(\lambda).$$

- ***Function hiding:*** *For a bit $b \in \{0,1\}$, we define the security experiment $\mathsf{ExptPRFE}_{\mathsf{FH}}[\lambda, \mathcal{A}, b]$ that describes the interaction between a challenger and a PPT adversary $\mathcal{A}$ exactly as $\mathsf{ExptPRFE}_{\mathsf{AS}}[\lambda, \mathcal{A}, b]$, except the challenge oracle is replaced with the following:*
  - ***Challenge query:*** *$\mathcal{A}$ submits a message μ^*, a revocation list $\mathcal{R}^* \subseteq \mathcal{ID}$, and two functions (f_0, f_1) such that for all triples (id, g, x) the adversary $\mathcal{A}$ submitted to $\mathcal{O}_{KG}$, it holds that $f_0(x) = f_1(x)$ or $\mathsf{id} \in \mathcal{R}^*$. The challenger replies with $\mathsf{ct}_b \leftarrow$ **Enc**$(\mathsf{pp}, \mathsf{msk}, \mu^*, \mathcal{R}^*, f_b)$ for $b \leftarrow \{0,1\}$. Note that the challenge query can be made only once.*

  At the end of the experiment, the adversary outputs a bit $b' \in \{0,1\}$, which is also the output of the experiment. We say $\Pi_{\mathbf{RPFE}}$ satisfies function hiding if for all PPT $\mathcal{A}$, we have

$$|\Pr[\mathsf{ExptPRFE}_{\mathsf{FH}}[\lambda, \mathcal{A}, 0] = 1] - \Pr[\mathsf{ExptPRFE}_{\mathsf{FH}}[\lambda, \mathcal{A}, 1] = 1]| = \mathrm{negl}(\lambda).$$

- ***Broadcast security:*** *For a bit $b \in \{0,1\}$, we define the security experiment $\mathsf{ExptPRFE}_{\mathsf{BS}}[\lambda, \mathcal{A}, b]$ that describes the interaction between a challenger and a PPT adversary $\mathcal{A}$ exactly as $\mathsf{ExptPRFE}_{\mathsf{AS}}[\lambda, \mathcal{A}, b]$, except the challenge oracle is replaced with the following:*
  - ***Challenge query:*** *$\mathcal{A}$ submits a message μ^* and a revocation list $\mathcal{R}^* \subseteq \mathcal{ID}$, the challenger computes $\mathsf{ct}_0 \leftarrow$ **Broadcast**$(\mathsf{pp}, \mu^*, \mathcal{R}^*)$, $\mathsf{ct}_1 \leftarrow$ **Enc**$(\mathsf{pp}, \mathsf{msk}, \mu^*, \mathcal{R}^*, f_{\mathsf{accept}})$ and returns ct_b for $b \leftarrow \{0,1\}$ to $\mathcal{A}$, where f_{accept} is the "always-accept" function that satisfies $f_{\mathsf{accept}} = 1$ for all $x \in \mathcal{X}$. Note that the challenge query can be made only once.*

  At the end of the experiment, the adversary outputs a bit $b' \in \{0,1\}$, which is also the output of the experiment. We say $\Pi_{\mathbf{RPFE}}$ satisfies broadcast security if for all PPT $\mathcal{A}$, we have

$$|\Pr[\mathsf{ExptPRFE}_{\mathsf{BS}}[\lambda, \mathcal{A}, 0] = 1] - \Pr[\mathsf{ExptPRFE}_{\mathsf{BS}}[\lambda, \mathcal{A}, 1] = 1]| = \mathrm{negl}(\lambda).$$

Definition 10 (Non-Adaptive q-Query Security). *For each of the security notions in Definition 9 (adaptive security, function hiding, and broadcast security), we define a notion of non-adaptive q-query security where the corresponding security notion only holds against all PPT adversaries that make at most $q \in \mathbb{N}$ queries to the encryption oracle, and moreover, all of the non-encryption queries occur before the encryption queries. Similar to [39], achieving this notion suffices for our trace-and-revoke functional encryption construction.*

4.2 Generic Construction of Secret-Key Revocable Predicate Functional Encryption with Broadcast

In this section, based on the framework of [39], we present a generic construction of secret-key RPFE with broadcast by combining a mixed FE scheme, an ABFE scheme, and a subset-cover set system.

Fix an identity space $\mathcal{ID} = \{0,1\}^n$, an attribute space $\mathcal{X}$, a message space $\mathcal{M}$, two function families $\mathcal{F} = \{f : \mathcal{X} \to \{0,1\}\}$ and $\mathcal{G}_\lambda = \{g : \mathcal{X}_\lambda \to \mathcal{Z}_\lambda\}$, where $n = n(\lambda)$ and $\mathcal{M} \subseteq \mathcal{X}_\lambda$.

- Let $[W]$ be the subset-cover set scheme for $\mathcal{ID} = \{0,1\}^n$. Let $\Pi_{\mathbf{SC}} = (\mathbf{Encode}, \mathbf{ComputeCover})$ be the subset-cover set scheme.

- Let $\Pi_{\mathbf{MFE}} = (\mathbf{MFE.PrmsGen}, \mathbf{MFE.MSKGen}, \mathbf{MFE.KeyGen}, \mathbf{MFE.PKEnc}, \mathbf{MFE.SKEnc}, \mathbf{MFE.Dec})$ be a mixed FE scheme with domain $\mathcal{X}$ and function family $\mathcal{F}$. For simplicity, we use $\tau = \tau(\lambda)$ to denote the randomness complexity of the master secret key generation algorithm **MFE.MSKGen**, $\mathcal{CT}$ denote the ciphertext space of $\Pi_{\mathbf{MFE}}$ (including **MFE.PKEnc** and **MFE.SKEnc**), and $\mathcal{SK}$ denote the secret key space of $\Pi_{\mathbf{MFE}}$. Moreover, we require that $\Pi_{\mathbf{MFE}}$ be sub-exponentially secure, and let $\epsilon > 0$ be a constant such that $2^{-\Omega(\lambda^\epsilon)}$ bounds the advantage of any efficient PPT adversary $\mathcal{A}$ for the security of $\Pi_{\mathbf{MFE}}$.
- For a secret key $\mathsf{mfe.sk} \in \mathcal{SK}$ and an index $i^* \in [W]$, we define the function $g_{\mathsf{mfe.sk},i^*} : \mathcal{CT} \times [W] \to \{0,1\}$ as:

$$g_{\mathsf{mfe.sk},i^*}(\mathsf{ct}, i) = \begin{cases} 1 & \mathbf{MFE.Dec}(\mathsf{mfe.sk}, \mathsf{ct}) = 1 \text{ and } i = i^*; \\ 0 & \text{otherwise.} \end{cases}$$

- Let $\Pi_{\mathbf{ABFE}} = (\mathbf{ABFE.Setup}, \mathbf{ABFE.KeyGen}, \mathbf{ABFE.Enc}, \mathbf{ABFE.Dec})$ be an attribute-based functional encryption scheme over message space $\mathcal{M}$, attribute space $\mathcal{X}$, function family $\mathcal{F}' = \{\mathsf{mfe.sk} \in \mathcal{SK}, i^* \in [W] : g_{\mathsf{mfe.sk},i^*}\}$ and $\mathcal{G}_\lambda$.
- Let $F : \mathcal{K} \times [W] \to \{0,1\}^\tau$ be a pseudorandom function.

Then, a secret-key RPFE with broadcast is constructed as follows:

- $\mathbf{Setup}(1^\lambda)$. On input a security parameter λ, set $\lambda' = \max\{\lambda, (\log W)^{2/\epsilon}\}$ and generate the mixed FE public parameters $\mathsf{mfe.pp} \leftarrow \mathbf{MFE.PrmsGen}(1^{\lambda'})$. Then, it generates the ABFE parameters $(\mathsf{abfe.pp}, \mathsf{abfe.msk}) \leftarrow \mathbf{ABFE.Setup}(1^\lambda)$, samples a PRF key $k \leftarrow \mathcal{K}$, and outputs

$$\mathsf{pp} = (\mathsf{mfe.pp}, \mathsf{abfe.pp}) \text{ and } \mathsf{msk} = (\mathsf{abfe.msk}, k).$$

- $\mathbf{KeyGen}(\mathsf{msk}, \mathsf{id}, g, x)$. On input msk, an identity $\mathsf{id} \in \mathcal{ID}$, a function $g \in \mathcal{G}_\lambda$, and an attribute $x \in \mathcal{X}$, it proceeds as follows:
  1. Compute a subset-cover encoding of the identity $\mathcal{I}_{\mathsf{id}} \leftarrow \mathbf{Encode}(\mathsf{id})$.
  2. For each index $i \in \mathcal{I}_{\mathsf{id}}$, compute $r_i = F(k, i)$, and produce a mixed FE master secret key $\mathsf{mfe.msk}_i \leftarrow \mathbf{MFE.MSKGen}(\mathsf{mfe.pp}; r_i)$ and a mixed FE decryption key $\mathsf{mfe.sk}_{i,x} \leftarrow \mathbf{MFE.KeyGen}(\mathsf{mfe.msk}_i, x)$.
  3. For each index $i \in \mathcal{I}_{\mathsf{id}}$, compute an ABFE decryption key with respect to the function $g_{\mathsf{mfe.sk}_{i,x},i}$ as follows: $\mathsf{abfe.sk}_{i,g,x} \leftarrow \mathbf{ABFE.KeyGen}(\mathsf{abfe.msk}, g, g_{\mathsf{mfe.sk}_{i,x},i})$.
  4. Output $\mathsf{sk}_{\mathsf{id},g,x} = \{(i, \mathsf{abfe.sk}_{i,g,x})\}_{i \in \mathcal{I}_{\mathsf{id}}}$.
- $\mathbf{Broadcast}(\mathsf{pp}, \mu, \mathcal{R})$: On input pp, a message $\mu \in \mathcal{M}$, and a revocation list $\mathcal{R} \subseteq \mathcal{ID}$, it proceeds as follows:
  1. Compute a cover $\mathcal{J}_{\mathcal{R}} \leftarrow \mathbf{ComputeCover}(\mathcal{R})$ for $\mathcal{ID} \backslash \mathcal{R}$.
  2. For each index $i \in \mathcal{J}_{\mathcal{R}}$, generate a mixed FE ciphertext $\mathsf{mfe.ct}_i \leftarrow \mathbf{MFE.PKEnc}(\mathsf{mfe.pp})$ and an ABFE ciphertext $\mathsf{abfe.ct}_i \leftarrow \mathbf{ABFE.Enc}(\mathsf{abfe.pp}, (\mathsf{mfe.ct}_i, i), \mu)$.
  3. Output $\mathsf{ct}_{\mu,\mathcal{R}} = \{(i, \mathsf{abfe.ct}_i)\}_{i \in \mathcal{J}_{\mathcal{R}}}$.
- $\mathbf{Enc}(\mathsf{pp}, \mathsf{msk}, \mu, \mathcal{R}, f)$. On input pp, msk, a message $\mu \in \mathcal{M}$, a revocation list $\mathcal{R} \subseteq \mathcal{ID}$, and a function $f \in \mathcal{F}$, it proceeds as follows:
  1. Compute a cover $\mathcal{J}_{\mathcal{R}} \leftarrow \mathbf{ComputeCover}(\mathcal{R})$ for $\mathcal{ID} \backslash \mathcal{R}$.
  2. For each index $i \in \mathcal{J}_{\mathcal{R}}$, compute $r_i = F(k, i)$, and produce a mixed FE master secret key $\mathsf{mfe.msk}_i \leftarrow \mathbf{MFE.MSKGen}(\mathsf{mfe.pp}; r_i)$ and a mixed FE ciphertext $\mathsf{mfe.ct}_i \leftarrow \mathbf{MFE.SKEnc}(\mathsf{mfe.msk}_i, f)$.
  3. For each index $i \in \mathcal{J}_{\mathcal{R}}$, generate an ABFE ciphertext $\mathsf{abfe.ct}_i \leftarrow \mathbf{ABFE.Enc}(\mathsf{abfe.pp}, (\mathsf{mfe.ct}_i, i), \mu)$.
  4. Output $\mathsf{ct}_{\mu,\mathcal{R},f} = \{(i, \mathsf{abfe.ct}_i)\}_{i \in \mathcal{J}_{\mathcal{R}}}$.

- **Dec**(sk, ct). Parse the secret key $\mathsf{sk} = \{(i, \mathsf{abfe.sk}_{i,g,x})\}_{i \in \mathcal{I}}$ and the ciphertext $\mathsf{ct} = \{(i, \mathsf{abfe.ct}_i)\}_{i \in \mathcal{J}}$, it proceeds as follows:
  1. Output $\perp$ if $\mathcal{I} \cap \mathcal{J} = \emptyset$.
  2. Otherwise, choose an arbitrary index $i \in \mathcal{I} \cap \mathcal{J}$ and output $g(\mu) \leftarrow$ **ABFE.Dec**($\mathsf{abfe.sk}_{i,g,x}, \mathsf{abfe.ct}_i$).

Correctness and security analysis. As an extension of the secret-key revocable predicate encryption with broadcast of [39], the correctness and security analysis of the above scheme is very similar to that of [39], as the main difference between the two schemes is that we use ABFE as the building block instead of ABE as used in [39].

Theorem 2 (Correctness). *Suppose that $\Pi_{\mathbf{SC}}$, $\Pi_{\mathbf{MFE}}$, and $\Pi_{\mathbf{ABFE}}$ are correct. Then, the above secret-key revocable predicate functional encryption with broadcast is correct.*

Proof. Given any message $\mu \in \mathcal{M}$, any attribute $x \in \mathcal{X}$, any function $f \in \mathcal{F}$ such that $f(x) = 1$, any identity $\mathsf{id} = (id_0, \ldots, id_n) \in \mathcal{ID}$, any revocation list $\mathcal{R} \subseteq \mathcal{ID}$ such that $\mathsf{id} \notin \mathcal{R}$, and any function $g \in \mathcal{G}_\lambda$. For $(\mathsf{pp}, \mathsf{msk}) = ((\mathsf{mfe.pp}, \mathsf{abfe.pp}), (\mathsf{abfe.msk}, k)) \leftarrow$ **Setup**(1^λ) and $\mathsf{sk}_{\mathsf{id},g,x} \leftarrow$ **KeyGen**($\mathsf{msk}, \mathsf{id}, g, x$) with $\mathsf{sk}_{\mathsf{id},g,x} = \{(i, \mathsf{abfe.sk}_{i,g,x})\}_{i \in \mathcal{I}_{\mathsf{id}}}$, where $\mathcal{I}_{\mathsf{id}} \leftarrow$ **Encode**(id), $\mathsf{abfe.sk}_{i,g,x}$ $\leftarrow$ **ABFE.KeyGen**($\mathsf{abfe.msk}, g, g_{\mathsf{mfe.sk}_{i,x},i}$), $r_i = F(k, i)$, $\mathsf{mfe.msk}_i \leftarrow$ **MFE.MSKGen**($\mathsf{mfe.pp}; r_i$) and $\mathsf{mfe.sk}_{i,x} \leftarrow$ **MFE.KeyGen**($\mathsf{mfe.msk}_i, x$). Then, consider the following correctness requirements:

- **Broadcast correctness.** For $\mathsf{ct}_{\mu,\mathcal{R}} \leftarrow$ **Broadcast**($\mathsf{pp}, \mu, \mathcal{R}$), we have $\mathsf{ct}_{\mu,\mathcal{R}} = \{(i, \mathsf{abfe.ct}_i)\}_{i \in \mathcal{J}_\mathcal{R}}$, where $\mathcal{J}_\mathcal{R} \leftarrow$ **ComputeCover**($\mathcal{R}$). Let $\mathsf{mfe.ct}_i \leftarrow$ **MFE.PKEnc**(mfe.pp) and $\mathsf{abfe.ct}_i \leftarrow$ **ABFE.Enc**($\mathsf{abfe.pp}, (\mathsf{mfe.ct}_i, i), \mu$) for $i \in \mathcal{J}_\mathcal{R}$. Since $\mathsf{id} \notin \mathcal{R}$, by correctness of $\Pi_{\mathbf{SC}}$, there exists an index $i \in \mathcal{I}_{\mathsf{id}} \cap \mathcal{J}$. Then, by correctness of $\Pi_{\mathbf{MFE}}$, we have **MFE.Dec**($\mathsf{mfe.sk}_{i,x}, \mathsf{mfe.ct}_i$) $= 1$ with overwhelming probability. This means that $g_{\mathsf{mfe.sk}_{i,x},i}(\mathsf{mfe.ct}_i, i) = 1$ and hence, by correctness of $\Pi_{\mathbf{ABFE}}$, we have **ABFE.Dec**($\mathsf{abfe.sk}_{i,g,x}, \mathsf{abfe.ct}_i$) $= g(\mu)$ with overwhelming probability.
- **Encryption correctness.** For $\mathsf{ct}_{\mu,\mathcal{R}} \leftarrow$ **Enc**($\mathsf{pp}, \mu, \mathcal{R}$), we have $\mathsf{ct}_{\mu,\mathcal{R},f} = \{(i, \mathsf{abfe.ct}_i)\}_{i \in \mathcal{J}_\mathcal{R}}$, where $\mathcal{J}_\mathcal{R} \leftarrow$ **ComputeCover**($\mathcal{R}$). Let $\mathsf{mfe.ct}_i \leftarrow$ **MFE.SKEnc**($\mathsf{mfe.msk}_i, f$) and $\mathsf{abfe.ct}_i \leftarrow$ **ABFE.Enc**($\mathsf{abfe.pp}, (\mathsf{mfe.ct}_i, i), \mu$) for $i \in \mathcal{J}_\mathcal{R}$. Since $\mathsf{id} \notin \mathcal{R}$, by correctness of $\Pi_{\mathbf{SC}}$, there exists an index $i \in \mathcal{I}_{\mathsf{id}} \cap \mathcal{J}$. Then, by correctness of $\Pi_{\mathbf{MFE}}$, we have **MFE.Dec**($\mathsf{mfe.sk}_{i,x}, \mathsf{mfe.ct}_i$) $=$ $f(x) = 1$ with overwhelming probability. This means that $g_{\mathsf{mfe.sk}_{i,x},i}(\mathsf{mfe.ct}_i, i) = 1$ and hence, by correctness of $\Pi_{\mathbf{ABFE}}$, we have **ABFE.Dec**($\mathsf{abfe.sk}_{i,g,x}, \mathsf{abfe.ct}_i$) $= g(\mu)$ with overwhelming probability.

Theorem 3 (Adaptive Security). *Suppose that $\Pi_{\mathbf{ABFE}}$ satisfies adaptive security. Then, the above secret-key revocable predicate functional encryption with broadcast satisfies adaptive security.*

Proof. The only difference between $\mathsf{ExptRPFE}_{\mathsf{AS}}[\lambda, \mathcal{A}, 0]$ and $\mathsf{ExptRPFE}_{\mathsf{AS}}[\lambda, \mathcal{A}, 1]$ is in how the challenge ciphertext ct_b is generated, where ct_b is generated by $\Pi_{\mathbf{ABFE}}$. We omit the detailed security proof, since it is not hard to see that this security proof is the same as that of Theorem 3.6 (Message Hiding) of [39], except that each relies on a different building block. More precisely, we need to prove that the ABFE ciphertexts in the $\mathsf{ExptRPFE}_{\mathsf{AS}}[\lambda, \mathcal{A}, 0]$ and $\mathsf{ExptRPFE}_{\mathsf{AS}}[\lambda, \mathcal{A}, 1]$ are computationally indistinguishable by the adaptive security of $\Pi_{\mathbf{ABFE}}$; in Theorem 3.6 of [39], they proved that the ABE ciphertexts in the $\mathsf{ExptRPE}_{\mathsf{MH}}[\lambda, \mathcal{A}, 0]$ and $\mathsf{ExptRPE}_{\mathsf{MH}}[\lambda, \mathcal{A}, 0]$ are computationally indistinguishable by the adaptive security of $\Pi_{\mathbf{ABE}}$.

Therefore, to prove Theorem 3, we only need to replace all $\Pi_{\mathbf{ABE}}$ of Theorem 3.6 of [39] with $\Pi_{\mathbf{ABFE}}$, and add a case of $(f(x) = 1, \mathsf{id} \notin \mathcal{R}, g(\mu_0) = g(\mu_1))$ in key query $\mathcal{O}_{\mathrm{KG}}$, which can be easily handled due to the adaptive security of $\Pi_{\mathbf{ABFE}}$. Overall, Theorem 3 follows by replacing all $\Pi_{\mathbf{ABE}}$ of Theorem 3.6 (Message Hiding) of [39] with $\Pi_{\mathbf{ABFE}}$.

Theorem 4 (Function Hiding). *Suppose that $\mathit{\Pi}_{\boldsymbol{MFE}}$ satisfies sub-exponential non-adaptive q-query (resp., adaptive) semantic security. Specifically, suppose that the advantage of any adversary running in time poly(λ) in the semantic security game is bounded by $2^{-\Omega(\lambda^{\epsilon})}$. In addition, suppose that $\mathit{\Pi}_{\boldsymbol{ABFE}}$ is secure and F is a secure PRF. Then, the above secret-key revocable predicate functional encryption with broadcast satisfies non-adaptive q-query (resp., adaptive) function hiding security.*

Proof. Similar to Theorem 3, Theorem 4 follows by replacing all $\mathit{\Pi}_{\mathbf{ABE}}$ of Theorem 3.7 (Function Hiding) of [39] with $\mathit{\Pi}_{\mathbf{ABFE}}$.

Theorem 5 (Broadcast Security). *Suppose that $\mathit{\Pi}_{\boldsymbol{MFE}}$ satisfies sub-exponential non-adaptive q-query (resp., adaptive) public/secret key indistinguishability. Specifically, suppose that the advantage of any adversary running in time poly(λ) in the public/secret key indistinguishability game is bounded by $2^{-\Omega(\lambda^{\epsilon})}$. Additionally, suppose that F is a secure PRF. Then, the above secret-key revocable predicate functional encryption with broadcast satisfies non-adaptive q-query (resp., adaptive) broadcast security.*

Proof. Since the broadcast security is independent of $\mathit{\Pi}_{\mathbf{ABFE}}$, Theorem 5 is identical to Theorem 3.8 (Broadcast Security) of [39].

4.3 Instantiation of Secret-Key Revocable Predicate Inner-Product Functional Encryption with Broadcast from LWE

In this section, based on the generic construction described in Section 4.2, we present an instantiation of secret-key revocable predicate inner-product functional encryption (RPIPFE) with broadcast by combining a mixed FE scheme, an ABIPFE scheme, and a subset-cover set system.

Definition 11 (Secret-Key RPIPFE with Broadcast). *A secret-key revocable predicate inner-product functional encryption with broadcast can be easily obtained from the secret-key revocable predicate functional encryption with broadcast (cf. Section 4.1). More precisely, its syntax and non-adaptive q-query security correspond to the corresponding definitions in Definitions 9 and 10, except with some substitutions:* 1) *let $\mathcal{G}_\lambda = \{\mathit{IP} : \mathcal{X}_\lambda \times \mathcal{Y}_\lambda \to \mathcal{Z}_\lambda\}$, and* 2) *replace all g with $y \in \mathcal{Y}_\lambda$, all μ with $x \in \mathcal{X}_\lambda$, and all $g(\mu)$ with $\mathit{IP}(x, y)$.*

Generic Construction of Secret-Key RPIPFE with Broadcast: We remark that the generic construction of the secret-key revocable predicate functional encryption with broadcast described in Section 4.2 implies a generic construction of the secret-key revocable predicate inner-product functional encryption with broadcast, using the same substitutions as Definition 11 and ABIPFE as defined in Definition 8. In addition, the non-adaptive q-query security of the secret-key revocable predicate inner-product functional encryption with broadcast follows from Theorems 3, 4 and 5.

Based on the above generic construction of the secret-key RPIPFE with broadcast, we obtain the following instantiation.

Instantiation from LWE: Fix an identity space $\mathcal{ID} = \{0,1\}^n$, an attribute space $\mathcal{X} = \{0,1\}^\ell$, three function families $\mathcal{F} = \{f : \{0,1\}^\ell \to \{0,1\}\}$, $\mathcal{F}' = \{\mathsf{mfe.sk} \in \mathcal{SK}, i^* \in [W] : g_{\mathsf{mfe.sk},i^*}\}$ and $\mathcal{G}_\lambda = \{\mathsf{IP} : \mathcal{X}_\lambda \times \mathcal{Y}_\lambda \to \mathcal{Z}_\lambda\}$, where $n = n(\lambda)$, $\ell = \ell(\lambda)$, $\mathcal{X}_\lambda = \{0, \ldots, P-1\}^m$, $\mathcal{Y}_\lambda = \{0, \ldots, V-1\}^m$, and $\mathcal{Z}_\gamma = \{0, \ldots, K-1\}$ for $K = mPV$. Then, we instantiate the construction of the secret-key RPIPFE with broadcast over the identity space $\mathcal{ID}$, attribute space $\mathcal{X}$, and function families $\mathcal{F}, \mathcal{G}_\lambda$, using the following building blocks:

- **Subset-cover set scheme over $\mathcal{ID}$**: using the subset-cover set system from Lemma 8.
- **Mixed FE scheme over $\{0,1\}^\ell$ and $\mathcal{F}$**: using the construction of Chen *et al.* [18].
- **ABIPFE over $\mathcal{F}'$ and $\mathcal{G}_\lambda$**: using the construction presented in Section 3.
- **Pseudorandom function**: using the construction of Kim [37].

5 Trace-and-Revoke Functional Encryption

In this section, we describe how to construct a trace-and-revoke function encryption scheme using a secret-key revocable predicate function encryption with broadcast. We begin by providing the formal definition of a trace-and-revoke functional encryption system, which can be seen as a direct generalization of the trace-and-revoke system.

5.1 Definition of Trace-and-Revoke Functional Encryption

We introduce the syntax of trace-and-revoke FE, which is the same as that of FE, except that it adds a revocation mechanism and a tracing algorithm. On the security side, compared with FE, trace-and-revoke IPFE has indistinguishability-based security and black-box traceability.

Fix an identity space $\mathcal{ID} = \{0,1\}^n$, a message space $\mathcal{M}$, and a function family $\mathcal{G}_\lambda = \{g : \mathcal{X}_\lambda \to \mathcal{Z}_\lambda\}$, where $n = n(\lambda)$ and $\mathcal{M} \subseteq \mathcal{X}_\lambda$. A trace-and-revoke FE scheme consists of a tuple of algorithms $(\mathbf{Setup}, \mathbf{KeyGen}, \mathbf{Enc}, \mathbf{Dec}, \mathbf{Trace})$, defined as follows:

- $\mathbf{Setup}(1^\lambda)$. On input a security parameter λ, output a public parameters pp and a master secret key msk.
- $\mathbf{KeyGen}(\mathsf{msk}, \mathsf{id}, g)$. On input msk, an identity $\mathsf{id} \in \mathcal{ID}$, and a function $g \in \mathcal{G}_\lambda$, output a secret key $\mathsf{sk}_{\mathsf{id},g}$.
- $\mathbf{Enc}(\mathsf{pp}, \mathcal{R}, \mu)$. On input pp, a revocation list $\mathcal{R} \subseteq \mathcal{ID}$, and a message $\mu \in \mathcal{M}$, output a ciphertext $\mathsf{ct}_{\mu,\mathcal{R}}$.
- $\mathbf{Dec}(\mathsf{sk}_{\mathsf{id},g}, \mathsf{ct}_{\mu,\mathcal{R}})$. On input $\mathsf{sk}_{\mathsf{id},g}$ and $\mathsf{ct}_{\mu,\mathcal{R}}$, output $\mu \in \mathcal{M}$ or $\perp$.
- $\mathbf{Trace}^{\mathcal{D}}(\mathsf{msk}, \mathcal{R}, \varepsilon, \mu_0, \mu_1)$. On input msk, $\mathcal{R} \subseteq \mathcal{ID}$, a non-negligible function $\varepsilon(\cdot)$ in λ and two different messages μ_0, μ_1, the tracing algorithm interacts with a ε-useful (see Definition 12) black-box distinguisher $\mathcal{D}$ and outputs an identity list $\mathcal{T} \subseteq \mathcal{ID}$ of malicious user(s) (note that $\mathcal{T}$ can be an empty list) or $\perp$. As shown in [23,30,48], this notion is stronger than the classical decryption black-box, where the latter is said to be "useful" if it can successfully decrypt (with a non-negligible probability) random messages that have been properly encrypted.

Definition 12 (ε-Useful Black-box Distinguisher [23]). *For a non-negligible function $\varepsilon(\cdot)$ in λ and a PPT algorithm $\mathcal{A}$, we say that a black-box distinguisher $\mathcal{D}$ is ε-useful, if we have*

$$\left| \Pr\left[\mathcal{D}(ct_b) = b : \begin{array}{c} (\mathcal{D}, \mathcal{R}, \mu_0, \mu_1) \leftarrow \mathcal{A}(pp, \{sk_{id,g}\}) \\ s.t.\ g(\mu_0) \neq g(\mu_1) \\ b \leftarrow \{0,1\}, ct_b \leftarrow \boldsymbol{Enc}(pp, \mathcal{R}, \mu_b) \end{array} \right] - \frac{1}{2} \right| \geq \varepsilon(\lambda),$$

where $(pp, msk) \leftarrow \boldsymbol{Setup}(1^\lambda)$ and $\{sk_{id,g}\} \leftarrow \boldsymbol{KeyGen}(msk, id, g)$ for $id \in \mathcal{ID}$ and $g \in \mathcal{G}_\lambda$.

Definition 13 (Correctness). *A trace-and-revoke FE scheme is correct if for any $\lambda \in \mathbb{N}$, any $g \in \mathcal{G}_\lambda$, any revocation list $\mathcal{R} \subseteq \mathcal{ID}$, and any $id \in \mathcal{ID}$ such that $id \notin \mathcal{R}$, we have*

$$\Pr\big[\mathbf{Dec}\big(sk_{id,g}, \mathbf{Enc}(pp, \mathcal{R}, \mu)\big) = g(\mu)\big] = 1 - \mathrm{negl}(\lambda),$$

where $(pp, msk) \leftarrow \boldsymbol{Setup}(1^\lambda)$ and $sk_{id,g} \leftarrow \boldsymbol{KeyGen}(msk, id, g)$.

Definition 14 (Adaptive Security). *We define adaptive security under chosen-plaintext attacks (A-IND-CPA security, for short) in the standard model, which is described by a security game between a PPT adversary $\mathcal{A}$ and a challenger, as follows:*

- ***Setup:*** *The challenger computes* $(\mathsf{pp}, \mathsf{msk}) \leftarrow \textbf{\textit{Setup}}(1^\lambda)$ *and returns* pp *to* $\mathcal{A}$.
- ***Key query:*** $\mathcal{A}$ *makes the following queries:*
  - ***Revoked user's key query*** $\mathcal{O}_{\boldsymbol{RUKQ}}$*: This query can be made only once.* $\mathcal{A}$ *submits a revocation list* $\mathcal{R}^* \subseteq \mathcal{ID}$ *and the corresponding functions* $g_{\mathsf{id}_j} \in \mathcal{G}_\lambda$ *for* $\mathsf{id}_j \in \mathcal{R}^*$*, the challenger responds with* $\mathsf{sk}_{\mathsf{id}_j, g_{\mathsf{id}_j}} \leftarrow \textbf{\textit{KeyGen}}(\mathsf{msk}, \mathsf{id}_j, g_{\mathsf{id}_j})$ *for all* $\mathsf{id}_j \in \mathcal{R}^*$.
  - ***Non-revoked user's key query*** $\mathcal{O}_{\boldsymbol{NRUKQ}}$*: Whenever* $\mathcal{A}$ *submits a pair* $(\mathsf{id}, g) \in \mathcal{ID} \times \mathcal{G}_\lambda$ *such that* $\mathsf{id} \notin \mathcal{R}^*$*, the challenger responds with* $\mathsf{sk}_{\mathsf{id},g} \leftarrow \textbf{\textit{KeyGen}}(\mathsf{msk}, \mathsf{id}, g)$.
- ***Challenge query:*** $\mathcal{A}$ *submits two messages* $\mu_0, \mu_1 \in \mathcal{M}$ *such that* $g(\mu_0) = g(\mu_1)$ *for all* $g \in \mathcal{G}_\lambda$ *the adversary submitted to* $\mathcal{O}_{\boldsymbol{NRUKQ}}$*, the challenger chooses a random bit* $b \leftarrow \{0, 1\}$ *and returns* $\mathsf{ct}_b \leftarrow \mathbf{Enc}(\mathsf{pp}, \mathcal{R}^*, \mu_b)$*. Note that this challenge query can be made only once.*
- ***Output:*** $\mathcal{A}$ *returns a bit* $b' \in \{0, 1\}$ *and wins if* $b' = b$.

The advantage of $\mathcal{A}$ *in winning the above game is defined as*

$$\mathrm{Adv}^{\text{A-IND-CPA}}_{\text{TR-FE},\mathcal{A}}(1^\lambda) = |\Pr[b' = b] - 1/2|.$$

We say that a trace-and-revoke FE scheme is **A-IND-CPA** *secure if for all PPT* $\mathcal{A}$*, the advantage* $\mathrm{Adv}^{\text{A-IND-CPA}}_{\text{TR-FE},\mathcal{A}}(1^\lambda)$ *is negligible.*

A weaker notion of adaptive security described above is called selective security against chosen-plaintext attacks (S-IND-CPA security, for short), where the adversary must announce the challenge messages (μ_0, μ_1) before seeing the public key.

Definition 15 (Black-Box Traceability). *Black-box traceability is described by a security game between a PPT adversary* $\mathcal{A}$ *and a challenger, as follows:*

- ***Setup:*** *The challenger computes* $(\mathsf{pp}, \mathsf{msk}) \leftarrow \textbf{\textit{Setup}}(1^\lambda)$ *and returns* pp *to* $\mathcal{A}$.
- ***Key query:*** $\mathcal{A}$ *makes the following queries:*
  - ***Revoked user's key query*** $\mathcal{O}_{\boldsymbol{RUKQ}}$**.** *This is the same as that of the A-IND-CPA security game described earlier.*
  - ***Non-revoked user's key query*** $\mathcal{O}_{\boldsymbol{NRUKQ}}$**.** *This is the same as that of the A-IND-CPA security game described earlier, except for brevity, we collect all pairs that were queried in* $\mathcal{O}_{\boldsymbol{NRUKQ}}$ *as* $\mathcal{C} = \{(\mathsf{id}, g) \mid (\mathsf{id}, g) \in \mathcal{ID} \times \mathcal{G}_\lambda\}$*, where all identities* $\mathsf{id} \in \mathcal{C}$ *are denoted by an identity list* $\mathcal{I} = \{\mathsf{id} \mid \mathsf{id} \in \mathcal{C}\}$.
- ***Black-box distinguisher generation:*** $\mathcal{A}$ *outputs a* ε*-useful black-box distinguisher* $\mathcal{D}$ *and two messages* $\mu_0, \mu_1 \in \mathcal{M}$ *such that* $g(\mu_0) \neq g(\mu_1)$ *for all functions* $g \in \mathcal{C}$.
- ***Output:*** *The challenger runs* $\mathbf{Trace}^{\mathcal{D}}(\mathsf{msk}, \mathcal{R}^*, \varepsilon, \mu_0, \mu_1)$ *and outputs an identity list* $\mathcal{T} \subseteq \mathcal{ID}$ *of malicious user(s).*

$\mathcal{A}$ *wins the above game if we have:*

1. *The provided black-box distinguisher* $\mathcal{D}$ *is indeed* ε*-useful, i.e., we have*

$$|\Pr[\mathcal{D}(\mathsf{ct}_b) = b] - \frac{1}{2}| \geq \varepsilon,$$

   where $\mathsf{ct}_b \leftarrow \textbf{\textit{Enc}}(\mathsf{pp}, \mathcal{R}^*, \mu_b)$ *and* $b \leftarrow \{0, 1\}$.
2. $\mathcal{T} = \emptyset$ *or* $\mathcal{T} \not\subseteq \mathcal{I}$.

We denote by $\mathrm{Adv}^{\mathrm{BT}}_{\mathrm{TR\text{-}FE},\mathcal{A}}$ *the advantage that* $\mathcal{A}$ *wins the above game, and we say that a trace-and-revoke FE scheme satisfies black-box traceability if for all PPT* $\mathcal{A}$*, the advantage* $\mathrm{Adv}^{\mathrm{BT}}_{\mathrm{TR\text{-}FE},\mathcal{A}}$ *is negligible for any* ε*-useful black-box distinguisher* $\mathcal{D}$*, where* ε *is non-negligible.*

Remark 2. A traceable IPFE scheme is simply a trace-and-revoke IPFE scheme without user revocation mechanism (i.e. let $\mathcal{R} = \emptyset$). In the traceable IPFE scheme of [23], the authors considered a weaker black-box traceability that they called one-target security, where the adversary are subject to the following constraints: (1) the adversary must announce one target $x^* \in \mathcal{X}$ before she sees the public key, (2) the adversary is only allowed to ask for secret keys for x^* (but for any index i), and (3) the adversary outputs a ε-useful black-box distinguisher $\mathcal{D}_{x^*}$ associated with x^*.

5.2 Construction of Trace-and-Revoke FE

Similar to the one in [39], our generic construction uses a secret-key RPFE scheme to embed an instance of the generalized jump-finding problem (cf. Definition 4), where the positions of the "jumps" correspond to non-revoked keys, and the tracing algorithm uses the generalized jump-finding algorithm (cf. Lemma 9) to identify traitors.

Fix an identity space $\mathcal{ID} = \{0,1\}^n$, a message space $\mathcal{M}$, and a function family $\mathcal{G}_\lambda = \{g : \mathcal{X}_\lambda \to \mathcal{Z}_\lambda\}$, where $n = n(\lambda)$ and $\mathcal{M} \subseteq \mathcal{X}_\lambda$.

- Let $\mathcal{ID}_0 = [2^{\ell+1}]$, and for any pair $(i,z) \in [n] \times [0, 2^{\ell+1}]$, define a function $f_{i,z} : \mathcal{ID}_0^n \to \{0,1\}$ that takes as input $\mathbf{y} = (y_1, \ldots, y_n)$, where $y_i \in \mathcal{ID}_0$ for $i \in [n]$, and outputs 1 if $y_i \leq z$ and 0 otherwise. It is not hard to see that $f_{i,z}(\mathbf{y}) = 0$ (called "all-zeros" function) for all $i \in [n]$ and $\mathbf{y} \in \mathcal{ID}_0^n$ when $z = 0$, and that $f_{i,z}(\mathbf{y}) = 1$ (called "all-ones" function) for all $i \in [n]$ and $\mathbf{y} \in \mathcal{ID}_0^n$ when $z = 2^{\ell+1}$.
- Let $\mathit{\Pi}_{\mathbf{RPFE}} = (\mathbf{RPFE.Setup}, \mathbf{RPFE.KeyGen}, \mathbf{RPFE.Broadcast}, \mathbf{RPFE.Enc}, \mathbf{RPFE.Dec})$ be a secret-key RPFE with broadcast over attribute space $\mathcal{ID}_0^n$, label space $[2^\ell]$, message space $\mathcal{M}$, and function space $\mathcal{F} = \{i \in [n], z \in [0, 2^{\ell+1}] : f_{i,z}\}$.
- Let $H : \mathcal{K} \times \mathcal{ID} \to [2^\ell]$ be a keyed collision-resistant hash function.

Then, a trace-and-revoke functional encryption scheme is constructed as follows:

- $\mathbf{Setup}(1^\lambda)$. On input a security parameter λ, choose $\mathsf{hk} \leftarrow \mathcal{K}$, run $(\mathsf{rpfe.pp}, \mathsf{rpfe.msk}) \leftarrow \mathbf{RPFE.Setup}(1^\lambda)$, and output
$$\mathsf{pp} = (\mathsf{hk}, \mathsf{rpfe.pp}) \quad \text{and} \quad \mathsf{msk} = (\mathsf{hk}, \mathsf{rpfe.msk}).$$
- $\mathbf{KeyGen}(\mathsf{msk}, \mathsf{id}, g)$. On input msk, an identity $\mathsf{id} = (id_0, \ldots, id_n) \in \mathcal{ID}$, and a function $g \in \mathcal{G}_\lambda$, it proceeds as follows:
  1. Compute $s_{\mathsf{id}} \leftarrow H(\mathsf{hk}, \mathsf{id})$ and let $\mathbf{y}_{\mathsf{id}} = (2s_{\mathsf{id}} - id_0, \ldots, 2s_{\mathsf{id}} - id_n) \in \mathcal{ID}_0^n$.
  2. Output $\mathsf{sk}_{\mathsf{id},g} \leftarrow \mathbf{RPFE.KeyGen}(\mathsf{rpfe.msk}, s_{\mathsf{id}}, g, \mathbf{y}_{\mathsf{id}})$.
- $\mathbf{Enc}(\mathsf{pp}, \mathcal{R}, \mu)$. On input pp, a revocation list $\mathcal{R} \subseteq \mathcal{ID}$, and a message $\mu \in \mathcal{M}$, it proceeds as follows:
  1. Compute $\mathcal{R}' = \{\mathsf{id} \in \mathcal{R} : H(\mathsf{hk}, \mathsf{id})\} \subseteq \{0,1\}^\ell$.
  2. Output $\mathsf{ct}_{\mu,\mathcal{R}} \leftarrow \mathbf{RPFE.Broadcast}(\mathsf{rpfe.pp}, \mathcal{R}', \mu)$.
- $\mathbf{Dec}(\mathsf{sk}_{\mathsf{id},g}, \mathsf{ct}_{\mu,\mathcal{R}})$. On input $\mathsf{sk}_{\mathsf{id},g}$ and $\mathsf{ct}_{\mu,\mathcal{R}}$, output $\mu \leftarrow \mathbf{RPFE.Dec}(\mathsf{sk}_{\mathsf{id},g}, \mathsf{ct}_{\mu,\mathcal{R}})$.
- $\mathbf{Trace}^{\mathcal{D}}(\mathsf{msk}, \mathcal{R}, \varepsilon, \mu_0, \mu_1)$. On input msk, $\mathcal{R} \subseteq \mathcal{ID}$, a non-negligible function $\varepsilon(\cdot)$ in λ and two different messages μ_0, μ_1, it proceeds as follows:

1. Build the list $\mathcal{R}' = \{\mathsf{id} \in \mathcal{R} : H(\mathsf{hk}, \mathsf{id})\} \subseteq \{0,1\}^{\ell}$.
2. Interact with a ε-useful black-box distinguisher $\mathcal{D}$ via the randomized oracle $\mathcal{Q}$ described in Fig. 1.
3. Let $q = 1$, set $\delta_q = \varepsilon/(9 + 4(\ell - 1)q)$, and compute $\mathcal{T}_q \leftarrow \mathbf{QTrace}^{\mathcal{Q}}(\lambda, 2^{\ell}, n, q, \delta_q, \varepsilon)$.
4. If $\mathcal{T}_q \neq \emptyset$, take any element $(s_{\mathsf{id}}, id_1, \ldots, id_n) \in \mathcal{T}_q$, and output $\mathsf{id} = (id_1, \ldots, id_n) \in \mathcal{ID}$. Otherwise, update $q \leftarrow 2q$ and repeat this procedure. [3]

```
On input $(i, z) \in [n] \times [0, 2^{\ell+1}]$:
  1. Choose a random bit $b \leftarrow \{0,1\}$ and generate a ciphertext
     $\mathsf{ct}_b \leftarrow \mathbf{RPFE.Enc}(\mathsf{rpfe.pp}, \mathsf{rpfe.msk}, \mathcal{R}', \mu_b, f_{i,z})$.
  2. Feed $\mathcal{D}$ with $\mathsf{ct}_b$ and obtain a binary value $b' \leftarrow \mathcal{D}(\mathsf{ct}_b)$.
  3. Output 1 if $b = b'$ and 0 otherwise.
```

Fig. 1: The randomized oracle $\mathcal{Q}$.

Correctness and security analysis. We show that the above trace-and-revoke functional encryption scheme satisfies correctness, semantic security, and black-box traceability. We remark that the analysis proceeds similarly to the corresponding analysis from [39].

Theorem 6 (Correctness). *Suppose that H and $\Pi_{\mathbf{RPFE}}$ are correct. Then, the above trace-and-revoke functional encryption scheme is correct.*

Proof. Given any message $\mu \in \mathcal{M}$, any identity $\mathsf{id} = (id_0, \ldots, id_n) \in \mathcal{ID}$, any function $g \in \mathcal{G}_\lambda$, and any revocation list $\mathcal{R} \subseteq \mathcal{ID}$ such that $\mathsf{id} \notin \mathcal{R}$. For $(\mathsf{pp}, \mathsf{msk}) = ((\mathsf{hk}, \mathsf{rpfe.pp}), (\mathsf{hk}, \mathsf{rpfe.msk})) \leftarrow \mathbf{Setup}(1^{\lambda})$, $\mathsf{sk}_{\mathsf{id},g} \leftarrow \mathbf{KeyGen}(\mathsf{msk}, \mathsf{id}, g)$, and $\mathsf{ct}_{\mu,\mathcal{R}} \leftarrow \mathbf{Enc}(\mathsf{pp}, \mathcal{R}, \mu)$, we know that $\mathsf{sk}_{\mathsf{id},g}$ was generated by $\mathbf{RPFE.KeyGen}(\mathsf{rpfe.msk}, s_{\mathsf{id}}, g, \mathbf{y}_{\mathsf{id}})$, where $s_{\mathsf{id}} \leftarrow H(\mathsf{hk}, \mathsf{id})$ and $\mathbf{y}_{\mathsf{id}} = (2s_{\mathsf{id}} - id_0, \ldots, 2s_{\mathsf{id}} - id_n) \in \mathcal{ID}_0^n$, and that $\mathsf{ct}_{\mu,\mathcal{R}}$ was generated by $\mathbf{RPFE.Broadcast}(\mathsf{rpfe.pp}, \mathcal{R}', \mu)$, where $\mathcal{R}' = \{\mathsf{id} \in \mathcal{R} : H(\mathsf{hk}, \mathsf{id})\} \subseteq \{0,1\}^{\ell}$. Since H is collision-resistant and $\mathsf{id} \notin \mathcal{R}$, we have $H(\mathsf{hk}, \mathsf{id}) \notin \mathcal{R}'$ with overwhelming probability. Then, the correctness holds due to the broadcast correctness of $\Pi_{\mathbf{RPFE}}$.

Theorem 7 (Adaptive Security). *Suppose that $\Pi_{\mathbf{RPFE}}$ satisfies adaptive security and broadcast security (without encryption queries), then the above trace-and-revoke functional encryption scheme is* **A-IND-CPA** *secure.*

Proof. The proof consists of the following hybrids, where the first hybrid corresponds to the real adaptive security game as defined in Definition 14. In the last hybrid, the adversary $\mathcal{A}$ has zero advantage due to the adaptive security and broadcast security of $\Pi_{\mathbf{RPFE}}$. Recall that the adaptive security of $\Pi_{\mathbf{RPFE}}$ (Theorem 3) can be achieved from message hiding of [39] by replacing all $\Pi_{\mathbf{ABE}}$ of Theorem 3.6 of [39] with $\Pi_{\mathbf{ABFE}}$, and the broadcast security of $\Pi_{\mathbf{RPFE}}$ (Theorem 5) is the same as that of [39]. Therefore, we omit the detailed security proof, as it is obvious that Theorem 7 follows by replacing $\Pi_{\mathbf{RPE}}$ of Theorem 4.6 (Semantic Security) of [39] with $\Pi_{\mathbf{RPFE}}$.

Theorem 8 (Black-Box Traceability). *Suppose that H is collision-resistant and $\Pi_{\mathbf{RPFE}}$ satisfies non-adaptive 1-query adaptive security, non-adaptive 1-query function hiding, and non-adaptive 1-query broadcast security, then the above trace-and-revoke functional encryption scheme satisfies black-box traceability. In particular, the tracing algorithm is efficient.*

[3] Similar to [39], the proof of Theorem 8 will show that this algorithm will terminate with overwhelming probability.

Proof. According to black-box traceability definition (i.e. Definition 15), at the beginning of the black-box traceability experiment, the challenger computes $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathbf{Setup}(1^\lambda)$, returns $\mathsf{pp} = (\mathsf{hk}, \mathsf{rpfe.pp})$ to $\mathcal{A}$ and keeps $\mathsf{msk} = (\mathsf{hk}, \mathsf{rpfe.msk})$ for itself. Let $\mathcal{R}^* \subseteq \mathcal{ID}$ be a revocation list submitted by $\mathcal{A}$ in $\mathcal{O}_{\mathrm{RUKQ}}$ and $\mathcal{C} = \{(\mathsf{id}, g) \mid (\mathsf{id}, g) \in \mathcal{ID} \times \mathcal{G}_\lambda\}$ the set queried by $\mathcal{A}$ in $\mathcal{O}_{\mathrm{NRUKQ}}$, where we let $\mathcal{I} = \{\mathsf{id} \mid \mathsf{id} \in \mathcal{C}\}$. At the end of the black-box traceability experiment, $\mathcal{A}$ outputs a ε-useful black-box distinguisher $\mathcal{D}$ and two messages $\mu_0^*, \mu_1^* \in \mathcal{M}$ such that $g(\mu_0^*) \neq g(\mu_1^*)$ for all functions $g \in \mathcal{C}$.

For each $\mathsf{id} \in \mathcal{I}$, let $s_{\mathsf{id}} \leftarrow H(\mathsf{hk}, \mathsf{id})$. By collision-resistance of H, all of the s_{id} will be distinct with overwhelming probability. Then, given the randomized oracle $\mathcal{Q}$, let $p_{i,z} := \Pr[\mathcal{Q}^{\mathcal{D}}(i, z) = 1]$ for any pair $(i, z) \in [n] \times [0, 2^{\ell+1}]$ and $p_{i,z,a} := \Pr[\mathsf{ct} \leftarrow \mathbf{RPFE.Enc}(\mathsf{rpfe.pp}, \mathsf{rpfe.msk}, \mathcal{R}'^*, \mu_a, f_{i,z}) : \mathcal{D}(\mathsf{ct}) = a]$ for $a \in \{0,1\}$, where $\mathcal{R}'^* = \{\mathsf{id} \in \mathcal{R}^* : H(\mathsf{hk}, \mathsf{id})\} \subseteq \{0,1\}^\ell$, we now show that the randomized oracle $\mathcal{Q}$ defines an instance of the $(2^\ell, n, |\mathcal{I}'|, \delta, \epsilon)$-generalized jump-finding problem for any $\delta = \varepsilon/(9+4(\ell-1)|\mathcal{I}'|)$ exactly as that of [39], where $\mathcal{I}' = \{\mathsf{id} \in \mathcal{I} : (s_{\mathsf{id}}, id_1, \ldots, id_n)\} \subseteq \{0,1\}^\ell$. We refer to Theorem 4.7 (Traceability) of [39] for detailed proof.

In conclusion, by Lemma 9 and the construction of the trace-and-revoke FE, the **Trace** algorithm will recover an element in $\mathcal{I}$ with overwhelming probability when executed on some $q > \log|\mathcal{I}'|$, so long as $\mathcal{T}_q \neq \emptyset$ and $\mathcal{T}_q \subseteq \mathcal{I}'$, where $\mathcal{T}_q \leftarrow \mathbf{QTrace}^{\mathcal{Q}}(\lambda, 2^\ell, n, q, \delta_q, \varepsilon)$.

5.3 Instantiation of Trace-and-Revoke IPFE with Broadcast from LWE

Definition 16 (Trace-and-Revoke IPFE). *With a few minor modification, we can get a trace-and-revoke IPFE from the trace-and-revoke FE described in Section 5.2. Specifically, its syntax, black-box distinguisher, correctness, adaptive security and black-box traceability correspond to the corresponding definitions in Definitions 12, 13, 14 and 15, except with some substitutions:* 1) *let* $\mathcal{G}_\lambda = \{\mathit{IP} : \mathcal{X}_\lambda \times \mathcal{Y}_\lambda \to \mathcal{Z}_\lambda\}$, *and* 2) *replace all* g *with* $y \in \mathcal{Y}_\lambda$, *all* μ *with* $x \in \mathcal{X}_\lambda$, *and all* $g(\mu)$ *with* $\mathit{IP}(x, y)$.

Generic Construction of Trace-and-Revoke IPFE: We remark that the generic construction of trace-and-revoke FE described in Section 5.2 implies a generic construction of trace-and-revoke IPFE, using the same substitutions as Definition 16 and the secret-key revocable predicate inner-product functional encryption with broadcast as defined in Definition 11. In addition, the correctness, adaptive security, and black-box traceability of the trace-and-revoke IPFE follows from Theorems 6, 7 and 8.

Based on the above generic construction of the trace-and-revoke IPFE, we obtain the following instantiation.

Instantiation from LWE: Fix an identity space $\mathcal{ID} = \{0,1\}^n$, an attribute space $\mathcal{X} = \{0,1\}^\ell$, two function families $\mathcal{F} = \{f : \{0,1\}^\ell \to \{0,1\}\}$ and $\mathcal{G}_\lambda = \{\mathsf{IP} : \mathcal{X}_\lambda \times \mathcal{Y}_\lambda \to \mathcal{Z}_\lambda\}$, where $n = n(\lambda)$, $\ell = \ell(\lambda)$, $\mathcal{X}_\lambda = \{0, \ldots, P-1\}^m$, $\mathcal{Y}_\lambda = \{0, \ldots, V-1\}^m$, and $\mathcal{Z}_\gamma = \{0, \ldots, K-1\}$ for $K = mPV$. Then, we instantiate the construction of trace-and-revoke IPFE with the LWE-based secret-key revocable predicate inner-product functional encryption with broadcast over identity space $\mathcal{ID}$, attribute space $\mathcal{X}$, and function families $\mathcal{F}, \mathcal{G}_\lambda$ as described in Section 4.3.

6 Conclusion and Future Work

In this paper, we considered a trace-and-revoke system on FE and proposed the first construction of trace-and-revoke FE. Compared to the previous trace-and-revoke IPFE schemes, our construction not only supports arbitrary identities, full collusion resistance, and full anonymity, but also admits trace-and-revoke FE for other restricted functionalities of practical interest (besides inner-product). We leave such instantiation as a future extension to this work. To instantiate our trace-and-revoke IPFE from LWE, we proposed the first LWE-based ABIPFE scheme that satisfies adaptive security (via a standard complexity leveraging argument), which provides an improvement over the previous work that satisfies somewhat weaker versions of adaptive security.

However, our construction does not support public traceability, as the tracing algorithm requires a secret key. Not surprisingly, this shortcoming is inherited from the identity-based trace-and-revoke scheme by Kim and J. Wu, which we extend here. In the future, it would be interesting to investigate how to upgrade our construction to support public traceability; this means that we will be constructing a public-key RPFE with broadcast (recall that the secret-key RPFE with broadcast is the building block of our construction). As the public-key RPFE with broadcast implies a public-key RPE with broadcast, we believe that if we want to achieve public traceability for our construction, the identity-based trace-and-revoke scheme by Kim and J. Wu would likely need to be improved, and the latter is currently an open problem.

References

1. M. Abdalla, F. Bourse, A. D. Caro, and D. Pointcheval. Simple functional encryption schemes for inner products. In J. Katz, editor, *Public-Key Cryptography - PKC 2015 - 18th IACR International Conference on Practice and Theory in Public-Key Cryptography, Gaithersburg, MD, USA, March 30 - April 1, 2015, Proceedings*, volume 9020 of *Lecture Notes in Computer Science*, pages 733–751. Springer, 2015.
2. M. Abdalla, D. Catalano, R. Gay, and B. Ursu. Inner-product functional encryption with fine-grained access control. *IACR Cryptol. ePrint Arch.*, 2020:577, 2020.
3. M. Abdalla, R. Gay, M. Raykova, and H. Wee. Multi-input inner-product functional encryption from pairings. In J. Coron and J. B. Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part I*, volume 10210 of *Lecture Notes in Computer Science*, pages 601–626, 2017.
4. S. Agrawal, S. Bhattacherjee, D. H. Phan, D. Stehlé, and S. Yamada. Efficient public trace and revoke from standard assumptions: Extended abstract. In B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, editors, *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 2277–2293. ACM, 2017.
5. S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*, pages 553–572, 2010.
6. S. Agrawal, B. Libert, and D. Stehlé. Fully secure functional encryption for inner products, from standard assumptions. In M. Robshaw and J. Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part III*, volume 9816 of *Lecture Notes in Computer Science*, pages 333–362. Springer, 2016.
7. M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 99–108, 1996.
8. C. E. Z. Baltico, D. Catalano, D. Fiore, and R. Gay. Practical functional encryption for quadratic functions with applications to predicate encryption. In J. Katz and H. Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part I*, volume 10401 of *Lecture Notes in Computer Science*, pages 67–98. Springer, 2017.
9. O. Billet and D. H. Phan. Efficient traitor tracing from collusion secure codes. In R. Safavi-Naini, editor, *Information Theoretic Security, Third International Conference, ICITS 2008, Calgary, Canada,*

August 10-13, 2008, Proceedings, volume 5155 of *Lecture Notes in Computer Science*, pages 171–182. Springer, 2008.

10. D. Boneh. The decision diffie-hellman problem. In J. Buhler, editor, *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings*, volume 1423 of *Lecture Notes in Computer Science*, pages 48–63. Springer, 1998.
11. D. Boneh and M. K. Franklin. An efficient public key traitor tracing scheme. In M. J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 338–353. Springer, 1999.
12. D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, and D. Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, pages 533–556, 2014.
13. D. Boneh, A. Sahai, and B. Waters. Fully collusion resistant traitor tracing with short ciphertexts and private keys. In S. Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 573–592. Springer, 2006.
14. D. Boneh, A. Sahai, and B. Waters. Functional encryption: Definitions and challenges. In Y. Ishai, editor, *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings*, volume 6597 of *Lecture Notes in Computer Science*, pages 253–273. Springer, 2011.
15. D. Boneh and B. Waters. A fully collusion resistant broadcast, trace, and revoke system. In A. Juels, R. N. Wright, and S. D. C. di Vimercati, editors, *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, Ioctober 30 - November 3, 2006*, pages 211–220. ACM, 2006.
16. Z. Brakerski and V. Vaikuntanathan. Circuit-abe from LWE: unbounded attributes and semi-adaptive security. In M. Robshaw and J. Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part III*, volume 9816 of *Lecture Notes in Computer Science*, pages 363–384. Springer, 2016.
17. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In H. Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*, pages 523–552. Springer, 2010.
18. Y. Chen, V. Vaikuntanathan, B. Waters, H. Wee, and D. Wichs. Traitor-tracing from LWE made simple and attribute-based. In A. Beimel and S. Dziembowski, editors, *Theory of Cryptography - 16th International Conference, TCC 2018, Panaji, India, November 11-14, 2018, Proceedings, Part II*, volume 11240 of *Lecture Notes in Computer Science*, pages 341–369. Springer, 2018.
19. Y. Chen, L. Zhang, and S. Yiu. Practical attribute based inner product functional encryption from simple assumptions. *IACR Cryptol. ePrint Arch.*, 2019:846, 2019.
20. B. Chor, A. Fiat, and M. Naor. Tracing traitors. In Y. Desmedt, editor, *Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings*, volume 839 of *Lecture Notes in Computer Science*, pages 257–270. Springer, 1994.
21. J. Chotard, E. D. Sans, R. Gay, D. H. Phan, and D. Pointcheval. Decentralized multi-client functional encryption for inner product. In T. Peyrin and S. D. Galbraith, editors, *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part II*, volume 11273 of *Lecture Notes in Computer Science*, pages 703–732. Springer, 2018.
22. P. Datta, T. Okamoto, and J. Tomida. Full-hiding (unbounded) multi-input inner product functional encryption from the k-linear assumption. In M. Abdalla and R. Dahab, editors, *Public-Key Cryptography - PKC 2018 - 21st IACR International Conference on Practice and Theory of Public-Key*

Cryptography, Rio de Janeiro, Brazil, March 25-29, 2018, Proceedings, Part II, volume 10770 of *Lecture Notes in Computer Science*, pages 245–277. Springer, 2018.

23. X. T. Do, D. H. Phan, and D. Pointcheval. Traceable inner product functional encryption. In S. Jarecki, editor, *Topics in Cryptology - CT-RSA 2020 - The Cryptographers' Track at the RSA Conference 2020, San Francisco, CA, USA, February 24-28, 2020, Proceedings*, volume 12006 of *Lecture Notes in Computer Science*, pages 564–585. Springer, 2020.
24. E. Gafni, J. Staddon, and Y. L. Yin. Efficient methods for integrating traceability and broadcast encryption. In M. J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 372–387. Springer, 1999.
25. S. D. Galbraith, K. Harrison, and D. Soldera. Implementing the tate pairing. In C. Fieker and D. R. Kohel, editors, *Algorithmic Number Theory, 5th International Symposium, ANTS-V, Sydney, Australia, July 7-12, 2002, Proceedings*, volume 2369 of *Lecture Notes in Computer Science*, pages 324–337. Springer, 2002.
26. S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 40–49. IEEE Computer Society, 2013.
27. S. Garg, C. Gentry, S. Halevi, and M. Zhandry. Functional encryption without obfuscation. In E. Kushilevitz and T. Malkin, editors, *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part II*, volume 9563 of *Lecture Notes in Computer Science*, pages 480–511. Springer, 2016.
28. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 197–206, 2008.
29. S. Gorbunov, V. Vaikuntanathan, and H. Wee. Predicate encryption for circuits from LWE. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, pages 503–523, 2015.
30. R. Goyal, V. Koppula, and B. Waters. Collusion resistant traitor tracing from learning with errors. In I. Diakonikolas, D. Kempe, and M. Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 660–670. ACM, 2018.
31. R. Goyal, V. Koppula, and B. Waters. New approaches to traitor tracing with embedded identities. In D. Hofheinz and A. Rosen, editors, *Theory of Cryptography - 17th International Conference, TCC 2019, Nuremberg, Germany, December 1-5, 2019, Proceedings, Part II*, volume 11892 of *Lecture Notes in Computer Science*, pages 149–179. Springer, 2019.
32. R. Goyal, W. Quach, B. Waters, and D. Wichs. Broadcast and trace with n$\hat{}\epsilon$ ciphertext size from standard assumptions. In A. Boldyreva and D. Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part III*, volume 11694 of *Lecture Notes in Computer Science*, pages 826–855. Springer, 2019.
33. R. Goyal, S. Vusirikala, and B. Waters. Collusion resistant broadcast and trace from positional witness encryption. In D. Lin and K. Sako, editors, *Public-Key Cryptography - PKC 2019 - 22nd IACR International Conference on Practice and Theory of Public-Key Cryptography, Beijing, China, April 14-17, 2019, Proceedings, Part II*, volume 11443 of *Lecture Notes in Computer Science*, pages 3–33. Springer, 2019.
34. D. Halevy and A. Shamir. The LSD broadcast encryption scheme. In M. Yung, editor, *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, volume 2442 of *Lecture Notes in Computer Science*, pages 47–60. Springer, 2002.
35. S. Katsumata and S. Yamada. Partitioning via non-linear polynomial functions: More compact ibes from ideal lattices and bilinear maps. In J. H. Cheon and T. Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II*, volume 10032 of *Lecture Notes in Computer Science*, pages 682–712, 2016.

36. C. H. Kim, Y. H. Hwang, and P. J. Lee. An efficient public key trace and revoke scheme secure against adaptive chosen ciphertext attack. In C. Laih, editor, *Advances in Cryptology - ASIACRYPT 2003, 9th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, November 30 - December 4, 2003, Proceedings*, volume 2894 of *Lecture Notes in Computer Science*, pages 359–373. Springer, 2003.
37. S. Kim. Key-homomorphic pseudorandom functions from LWE with a small modulus. *IACR Cryptol. ePrint Arch.*, page 233, 2020.
38. S. Kim, K. Lewi, A. Mandal, H. Montgomery, A. Roy, and D. J. Wu. Function-hiding inner product encryption is practical. In D. Catalano and R. D. Prisco, editors, *Security and Cryptography for Networks - 11th International Conference, SCN 2018, Amalfi, Italy, September 5-7, 2018, Proceedings*, volume 11035 of *Lecture Notes in Computer Science*, pages 544–562. Springer, 2018.
39. S. Kim and D. J. Wu. Collusion resistant trace-and-revoke for arbitrary identities from standard assumptions. In S. Moriai and H. Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part II*, volume 12492 of *Lecture Notes in Computer Science*, pages 66–97. Springer, 2020.
40. Q. Lai, F. Liu, and Z. Wang. New lattice two-stage sampling technique and its applications to functional encryption - stronger security and smaller ciphertexts. In A. Canteaut and F. Standaert, editors, *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part I*, volume 12696 of *Lecture Notes in Computer Science*, pages 498–527. Springer, 2021.
41. B. Libert and R. Titiu. Multi-client functional encryption for linear functions in the standard model from LWE. In S. D. Galbraith and S. Moriai, editors, *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part III*, volume 11923 of *Lecture Notes in Computer Science*, pages 520–551. Springer, 2019.
42. F. Luo and S. Al-Kuwari. Generic construction of black-box traceable attribute-based encryption. *IEEE Transactions on Cloud Computing*, 2021.
43. F. Luo, S. Al-Kuwari, H. Wang, and W. Han. Generic construction of trace-and-revoke inner product functional encryption. *To appear at ESORICS 2022*, 2022.
44. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, pages 700–718, 2012.
45. D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. In *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*, pages 372–381. IEEE Computer Society, 2004.
46. D. Naor, M. Naor, and J. Lotspiech. Revocation and tracing schemes for stateless receivers. In J. Kilian, editor, *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, volume 2139 of *Lecture Notes in Computer Science*, pages 41–62. Springer, 2001.
47. M. Naor and B. Pinkas. Efficient trace and revoke schemes. In Y. Frankel, editor, *Financial Cryptography, 4th International Conference, FC 2000 Anguilla, British West Indies, February 20-24, 2000, Proceedings*, volume 1962 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2000.
48. R. Nishimaki, D. Wichs, and M. Zhandry. Anonymous traitor tracing: How to embed arbitrary information in a key. In M. Fischlin and J. Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 388–419. Springer, 2016.
49. T. Pal and R. Dutta. Attribute-based access control for inner product functional encryption from LWE. In P. Longa and C. Ràfols, editors, *Progress in Cryptology - LATINCRYPT 2021 - 7th International Conference on Cryptology and Information Security in Latin America, Bogotá, Colombia, October 6-8, 2021, Proceedings*, volume 12912 of *Lecture Notes in Computer Science*, pages 127–148. Springer, 2021.

50. C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In M. Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 333–342. ACM, 2009.
51. C. Peikert. A decade of lattice cryptography. *Foundations and Trends in Theoretical Computer Science*, 10(4):283–424, 2016.
52. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In H. N. Gabow and R. Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 84–93. ACM, 2005.
53. A. Sahai and B. Waters. Fuzzy identity-based encryption. In R. Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473. Springer, 2005.
54. V. Shoup. *A computational introduction to number theory and algebra.* Cambridge University Press, 2006.
55. J. Staddon, D. R. Stinson, and R. Wei. Combinatorial properties of frameproof and traceability codes. *IEEE Trans. Inf. Theory*, 47(3):1042–1049, 2001.
56. D. R. Stinson and R. Wei. Combinatorial properties and constructions of traceability schemes and frameproof codes. *SIAM J. Discret. Math.*, 11(1):41–53, 1998.
57. D. R. Stinson and R. Wei. Key preassigned traceability schemes for broadcast encryption. In S. E. Tavares and H. Meijer, editors, *Selected Areas in Cryptography '98, SAC'98, Kingston, Ontario, Canada, August 17-18, 1998, Proceedings*, volume 1556 of *Lecture Notes in Computer Science*, pages 144–156. Springer, 1998.
58. J. Tomida. Tightly secure inner product functional encryption: Multi-input and function-hiding constructions. In S. D. Galbraith and S. Moriai, editors, *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part III*, volume 11923 of *Lecture Notes in Computer Science*, pages 459–488. Springer, 2019.
59. Z. Wang, X. Fan, and F. Liu. FE for inner products and its application to decentralized ABE. In D. Lin and K. Sako, editors, *Public-Key Cryptography - PKC 2019 - 22nd IACR International Conference on Practice and Theory of Public-Key Cryptography, Beijing, China, April 14-17, 2019, Proceedings, Part II*, volume 11443 of *Lecture Notes in Computer Science*, pages 97–127. Springer, 2019.
60. B. Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, editors, *Public Key Cryptography - PKC 2011 - 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, March 6-9, 2011. Proceedings*, volume 6571 of *Lecture Notes in Computer Science*, pages 53–70. Springer, 2011.

A Inner-Product Functional Encryption

We first give the syntax and security definition of inner-product function encryption. Then, we review the LWE-based inner-product function encryption scheme of Agrawal *et al.* [6], which we denote as ALS.

Definition 17 (Inner-Product Function Encryption). *For any inner-product function $\mathsf{IP}: \mathcal{X}_\lambda \times \mathcal{Y}_\lambda \rightarrow \mathcal{Z}_\lambda$, an inner-product function encryption scheme consists of a tuple of algorithms $\Pi_{\boldsymbol{IPFE}} = (\boldsymbol{Setup}, \boldsymbol{KeyGen}, \boldsymbol{Enc}, \boldsymbol{Dec})$, defined as the following:*

- ***Setup***(1^λ)*. Take as input a security parameter λ, output a master public/secret key pair (mpk, msk).*
- ***KeyGen***(msk, y)*. Take as input msk and $y \in \mathcal{Y}_\lambda$, output a secret key sk_y.*
- ***Enc***(mpk, x)*. Take as input mpk and message $x \in \mathcal{X}_\lambda$, output a ciphertext ct.*
- ***Dec***(sk_y, ct)*. Take as input sk_y and ct, output $z \in \mathcal{Z}_\lambda$ or $\perp$.*

Correctness. Given $(mpk, msk) \leftarrow \mathbf{Setup}(1^\lambda)$, $sk_y \leftarrow \mathbf{KeyGen}(msk, y)$ for any $y \in \mathcal{Y}_\lambda$, and message $x \in \mathcal{X}_\lambda$, we have $\Pr[\mathbf{Dec}\big(sk_y, \mathbf{Enc}(mpk, x)\big) = \mathsf{IP}(x, y)] = 1 - \mathrm{negl}(\lambda)$.

Definition 18 (Security for $\Pi_{\mathbf{IPFE}}$). *Here, we consider the indistinguishability security also called security against adaptive chosen-plaintext attacks (AD-CPA). The indistinguishability security experiment between a PPT adversary $\mathcal{A}$ and a challenger is described as the following:*

Setup: *The challenger runs $(mpk, msk) \leftarrow$ **Setup**(1^λ) and gives mpk to $\mathcal{A}$.*

Key generation query: *Whenever $\mathcal{A}$ submits $y \in \mathcal{Y}_\lambda$, the challenger responds with $sk_y \leftarrow$ **KeyGen**(msk, y) and stores sk_y in response to multiple secret key queries on the same y that $\mathcal{A}$ may make in the future.*

Challenge: *$\mathcal{A}$ submits two messages $x_0, x_1 \in \mathcal{X}_\lambda$ such that $\mathsf{IP}(x_0, y) = \mathsf{IP}(x_1, y)$ for all y the adversary submitted to the key generation query, the challenger chooses a uniformly random bit $b \leftarrow \{0, 1\}$ and returns $ct_b \leftarrow$ **Enc**(mpk, x_b). Note that the challenge oracle can be made only once.*

Output: *Finally, $\mathcal{A}$ returns a bit $b' \in \{0, 1\}$, which is also the output of the experiment.*

The advantage of $\mathcal{A}$ in winning the above experiment is defined as: $\mathrm{Adv}^{\text{AD-CPA}}_{\text{IPFE},\mathcal{A}} := |\Pr[b = b'] - 1/2|$.

A natural weaker notion of AD-CPA is security against selective chosen-plaintext attacks, where $\mathcal{A}$ must announce the two messages $x_0, x_1 \in \mathcal{X}_\lambda$ before she receives the master public key and makes any query. We say that $\Pi_{\mathbf{IPFE}}$ is AD-CPA secure if for all PPT $\mathcal{A}$ the advantage $\mathrm{Adv}^{\text{AD-CPA}}_{\text{IPFE},\mathcal{A}}$ is negligible.

ALS scheme [6]. Wang *et al.* [59] used the rerandomization technique [35] to simplify the security proof of the ALS scheme and improve its parameters. The LWE-based identity-based functional encryption of Abdalla *et al.* in [2] also relies on the rerandomization technique. Here, we review the variant of ALS scheme presented in [2], as identity-based and attribute-based systems rely on some of the same algorithms. Given $\mathcal{X}_\lambda = \{0, \ldots, P-1\}^\eta$, $\mathcal{Y}_\lambda = \{0, \ldots, V-1\}^\eta$, and $\mathcal{Z}_\lambda = \{0, \ldots, K-1\}$, where $K = \eta PV$, the ALS scheme is built as the following:

- **Setup**(1^λ). Sample matrices $\mathbf{A}_{\text{ALS}} \leftarrow \mathbb{Z}_q^{n\times m}, \mathbf{Z} \leftarrow \mathcal{D}_{\mathbb{Z}^{m\times\eta},\rho}$. Set $\mathbf{U}_{\text{ALS}} = \mathbf{A}_{\text{ALS}}\mathbf{Z}$. Output $mpk := (\mathbf{A}_{\text{ALS}}, \mathbf{U}_{\text{ALS}})$ and $msk := (\mathbf{Z})$.
- **KeyGen**$(msk, \mathbf{y})$. Take as input msk and a vector $\mathbf{y} \in \mathcal{Y}_\lambda$, output $sk_{\mathbf{y}} := (\mathbf{y}, \mathbf{Z}\cdot\mathbf{y})$.
- **Enc**$(mpk, \mathbf{x})$. Take as input msk and a vector $\mathbf{x} \in \mathcal{X}_\lambda$. Choose $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, two error $\mathbf{e}_0 \leftarrow \mathcal{D}_{\mathbb{Z}^m,\sigma}, \mathbf{e}_1 \leftarrow \mathcal{D}_{\mathbb{Z}^\eta,\sigma}$. Output a ciphertext
$$\mathbf{ct} = (\mathbf{c}_1^{\text{ALS}}, \mathbf{c}_2^{\text{ALS}}) = (\mathbf{A}_{\text{ALS}}^T\mathbf{s} + \mathbf{e}_0, \mathbf{U}_{\text{ALS}}^T\mathbf{s} + \mathbf{e}_1 + \lfloor q/K \rfloor \cdot \mathbf{x}).$$
- **Dec**$(sk_{\mathbf{y}}, \mathbf{ct})$. Parse $\mathbf{ct} = (\mathbf{c}_1^{\text{ALS}}, \mathbf{c}_2^{\text{ALS}})$. Compute $\mu' = \mathbf{y}^T\mathbf{c}_2^{\text{ALS}} - (\mathbf{Z}\cdot\mathbf{y})^T\mathbf{c}_1^{\text{ALS}} \pmod q$. Output the value $\mu \in \{-K+1, \ldots, K-1\}$ that minimizes $|\lfloor q/K \rfloor \cdot \mu - \mu'|$.

Parameters and Correctness. For correctness and security, they set the parameters as follows: $\lambda = n$, $\alpha \leq \sigma/(2C\alpha q(\sqrt{m} + \sqrt{n} + \sqrt{\eta}))$ for some constant $C \approx 1/\sqrt{2\pi}$, $\sigma \geq \omega(\sqrt{\log n})$, $\rho \geq \omega(\sqrt{\log n})$, $m = 2n\log q$, and $q \geq 4K(\eta V\omega(\sqrt{\log n}) + \eta Vm\omega(\log n))$ such that $\alpha q \geq \Omega(\sqrt{n})$. Based on the choice of parameters, we have

$$\begin{aligned}\mu' &= \mathbf{y}^T(\mathbf{U}_{\text{ALS}}^T\mathbf{s} + \mathbf{e}_1 + \lfloor q/K \rfloor \cdot \mathbf{x}) - (\mathbf{Z}\cdot\mathbf{y})^T(\mathbf{A}_{\text{ALS}}^T\mathbf{s} + \mathbf{e}_0)\\ &= \lfloor q/K \rfloor \cdot \mathbf{y}^T\mathbf{x} + \mathbf{y}^T\mathbf{e}_1 - (\mathbf{Z}\cdot\mathbf{y})^T\mathbf{e}_0,\end{aligned}$$

where we have $|\mathbf{y}^T\mathbf{e}_1 - (\mathbf{Z}\cdot\mathbf{y})^T\mathbf{e}_0| \leq \eta V\omega(\sqrt{\log n}) + \eta Vm\omega(\log n) \leq \lfloor q/K \rfloor/4$ with probability $1 - n^{-\omega(1)}$.

Theorem 9 ([6,59]). *Given parameters $\lambda, n, m, \alpha, \sigma, \rho, q$ as described above, the ALS scheme is AD-CPA secure, assuming the hardness of the* $\mathrm{LWE}_{n,m,q,\alpha}$ *problem.*

B Proof of Theorem 1

The proof consists of the following hybrids, where the first hybrid corresponds to the real security game as defined in Definition 8. In the last hybrid, the adversary $\mathcal{A}$ has zero advantage due to the AD-CPA security of the ALS scheme. In the following, we build these hybrids to prove that $\mathcal{A}$ has negligible advantage in winning the original selective security game.

• **Hybrid 0**: This hybrid corresponds to the real security game.

• **Hybrid 1**: This hybrid is identical to **Hybrid** 0, with the only difference being that we change how $\mathbf{U}$ is generated. In this hybrid, the challenger samples $\mathbf{X} \leftarrow \mathcal{D}_{\mathbb{Z}^{m \times m}, \sqrt{\gamma^2+s^2}}$ and sets $\mathbf{U} = \mathbf{AX}$.

• **Hybrid 2**: This hybrid is identical to **Hybrid** 1, with the only difference being that we change how the public matrices $(\mathbf{B}_1, \ldots, \mathbf{B}_\theta)$ are generated. Specifically, in this hybrid, upon receiving the target attribute $\mathsf{att}^* \in \{0,1\}^\theta$, the challenger samples θ matrices $\mathbf{S}_1^*, \ldots, \mathbf{S}_\theta^* \leftarrow \{-1,1\}^{m \times m}$ and computes $\mathbf{B}_i = \mathbf{A}\mathbf{S}_i^* - \mathsf{att}_i^*\mathbf{G}$ for $i \in [\theta]$.

• **Hybrid 3**: This hybrid is identical to **Hybrid** 2, with the only difference being that we change how $\mathbf{A}$ is generated. In this hybrid, the challenger samples $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$. The construction of the public matrices $(\mathbf{B}_1, \ldots, \mathbf{B}_\theta)$ remains as in **Hybrid** 2, that is, $\mathbf{B}_i = \mathbf{A}\mathbf{S}_i^* - \mathsf{att}_i^*\mathbf{G}$ for $i \in [\theta]$. In this hybrid, the challenger has no trapdoor of $\Lambda_q^\perp(\mathbf{A})$, but it can handle all $\mathcal{O}_{\mathbf{KG}}$, as follows.

First, by Definition 8, for any pair $(f, \mathbf{y}) \in \mathcal{F} \times \mathcal{Y}_\lambda$ submitted by $\mathcal{A}$ we consider the following two cases: 1) $f(\mathsf{att}^*) = 1$, and 2) $(f(\mathsf{att}^*) = 0, \langle \mathbf{x}_0, \mathbf{y} \rangle = \langle \mathbf{x}_1, \mathbf{y} \rangle)$, where $\mathbf{x}_0, \mathbf{x}_1$ are challenge vectors. The challenger generates a secret key $sk_{f,\mathbf{y}}$ for $(f, \mathbf{y})$ as follows:

1. **Case 1**: $f(\mathsf{att}^*) = 1$.
   a) Compute $\mathbf{B}_f = \mathbf{Eval}_{pk}\big(f, (\mathbf{B}_1, \ldots, \mathbf{B}_\theta)\big)$, choose $\mathbf{W} \leftarrow \mathcal{D}_{\mathbb{Z}^{m \times m}, \gamma}$ and let $\mathbf{D} = \mathbf{U} - \mathbf{AW}$.
   b) Run $\mathbf{S}_f^* \leftarrow \mathbf{Eval}_{sim}\big(f, \mathbf{A}, \{(\mathbf{S}_i^*, \mathsf{att}_i^*)\}_{i \in [\theta]}\big)$ such that $\mathbf{B}_f = \mathbf{A}\mathbf{S}_f^* - f(\mathsf{att}^*)\mathbf{G} = \mathbf{A}\mathbf{S}_f^* - \mathbf{G}$. By definition of $\mathbf{Eval}_{sim}$ algorithm of Lemma 7, we have $||\mathbf{S}_f^*||_2 < (m+1)^{d+1}$.
   c) Finally, run $\begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \end{bmatrix} \leftarrow \mathbf{SampleRight}(\mathbf{A}, \mathbf{G}, \mathbf{S}_f^*, \mathbf{T}_\mathbf{G}, \mathbf{D}, s)$ to produce a low-norm matrix $\begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \end{bmatrix} \in \mathbb{Z}^{2m \times m}$ such that $(\mathbf{A} | \mathbf{A}\mathbf{S}_f^* - \mathbf{G}) \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \end{bmatrix} = \mathbf{D}$. Let $\mathbf{R}_f = \begin{bmatrix} \mathbf{R}_1 + \mathbf{W} \\ \mathbf{R}_2 \end{bmatrix}$ and $sk_{f,\mathbf{y}} := (\mathbf{y}, \mathbf{R}_f \cdot \mathbf{y})$.
2. **Case 2**: $f(\mathsf{att}^*) = 0$ and $\langle \mathbf{x}_0, \mathbf{y} \rangle = \langle \mathbf{x}_1, \mathbf{y} \rangle$.
   In this case, we cannot use the public trapdoor $\mathbf{T}_\mathbf{G}$ to generate the secret key. Instead, since $\mathbf{U} = \mathbf{AX}$ for $\mathbf{X} \leftarrow \mathcal{D}_{\mathbb{Z}^{m \times m}, \sqrt{\gamma^2+s^2}}$, the challenger proceeds as the following:
   a) Compute $\mathbf{B}_f = \mathbf{Eval}_{pk}\big(f, (\mathbf{B}_1, \ldots, \mathbf{B}_\theta)\big)$.
   b) Run $\mathbf{S}_f^* \leftarrow \mathbf{Eval}_{sim}\big(f, \mathbf{A}, \{(\mathbf{S}_i^*, \mathsf{att}_i^*)\}_{i \in [\theta]}\big)$ such that $\mathbf{B}_f = \mathbf{A}\mathbf{S}_f^* - f(\mathsf{att}^*)\mathbf{G} = \mathbf{A}\mathbf{S}_f^*$. By definition of $\mathbf{Eval}_{sim}$ algorithm of Lemma 7, we have $||\mathbf{S}_f^*||_2 < (m+1)^{d+1}$.
   c) Finally, choose $\mathbf{R}_2 \leftarrow \mathcal{D}_{\mathbb{Z}^{m \times m}, s}$ and let $\mathbf{R}_f = \begin{bmatrix} \mathbf{X} - \mathbf{S}_f^*\mathbf{R}_2 \\ \mathbf{R}_2 \end{bmatrix}$. Then, let $sk_{f,\mathbf{y}} := (\mathbf{y}, \mathbf{R}_f \cdot \mathbf{y})$.

The challenger armed with the above ability responds all key queries raised by $\mathcal{A}$ as the following:

- **Key query** $\mathcal{O}_{\mathbf{KG}}$: Whenever $\mathcal{A}$ submits $(f, \mathbf{y}) \in \mathcal{F} \times \mathcal{Y}_\lambda$, the challenger produces a low-norm matrix $\mathbf{R}_f \in \mathbb{Z}_q^{2m \times m}$ as described above, returns $sk_{f,\mathbf{y}} = (\mathbf{y}, \mathbf{R}_f \cdot \mathbf{y})$ to $\mathcal{A}$, and stores $sk_{f,\mathbf{y}}$ in response to multiple key queries on the same $(f, \mathbf{y})$ that $\mathcal{A}$ may make in the future.

• **Hybrid** 4: This hybrid is identical to **Hybrid** 3, with the only difference being that we change how $\mathbf{A}, \mathbf{U}$ and the challenge ciphertext are generated. In this final hybrid, similar to the security proof of the LWE-based identity-based functional encryption of Abdalla *et al.* [2], we use the AD-CPA security of the ALS scheme (see Appendix A) to argue indistinguishability of ciphertexts. In this hybrid, the challenger interacts with the AD-CPA challenger of the ALS scheme. We describe the challenger's behaviour as follow:

Setup: Upon receiving the public keys $(\mathbf{A}_{\text{ALS}}, \mathbf{U}_{\text{ALS}})$ from the AD-CPA challenger of the ALS, the challenger sets $\mathbf{A} := \mathbf{A}_{\text{ALS}}, \mathbf{U} := \mathbf{U}_{\text{ALS}}$.

Key query: Consider the following two cases:

1. **Case 1**: Whenever $\mathcal{A}$ asks $\mathcal{O}_{\text{KG}}$ for any $(f, \mathbf{y})$, where $f(\mathsf{att}^*) = 1$, the challenger answers as in **Hybrid** 3.
2. **Case 2**: Whenever $\mathcal{A}$ asks $\mathcal{O}_{\text{KG}}$ for any $(f, \mathbf{y})$, where $f(\mathsf{att}^*) = 0$ and $\langle \mathbf{x}_0, \mathbf{y} \rangle = \langle \mathbf{x}_1, \mathbf{y} \rangle$, the challenger asks the AD-CPA challenger of the ALS for the secret key query on $\mathbf{y}$ and obtains $sk_{\mathbf{y}}$. Then, the challenger chooses $\mathbf{R}_2 \leftarrow \mathcal{D}_{\mathbb{Z}^{m \times m}, s}$ and let $\mathbf{R}_{f,\mathbf{y}} = \begin{bmatrix} sk_{\mathbf{y}} - \mathbf{S}_f^* \mathbf{R}_2 \cdot \mathbf{y} \\ \mathbf{R}_2 \cdot \mathbf{y} \end{bmatrix}$, where recall that $\mathbf{S}_f^* \leftarrow \mathbf{Eval}_{sim}\big(f, \mathbf{A}, \{(\mathbf{S}_i^*, \mathsf{att}_i^*)\}_{i \in [\theta]}\big)$. Finally, return $sk_{f,\mathbf{y}} := (\mathbf{y}, \mathbf{R}_{f,\mathbf{y}})$.

Challenge query: Upon receiving two messages $\mathbf{x}_0, \mathbf{x}_1 \in \mathcal{X}_\lambda$, the challenger forwards them to the AD-CPA challenger of the ALS and receives $(\mathbf{c}_1^{\text{ALS}}, \mathbf{c}_2^{\text{ALS}})$. Then, the challenger computes and returns:

$$\begin{aligned} \mathbf{c}_1 &= (\mathbf{I}_m \mid \mathbf{S}_1^* \mid \cdots \mid \mathbf{S}_\theta^*)^T \cdot \mathbf{c}_1^{\text{ALS}}, \\ \mathbf{c}_2 &= \mathbf{c}_2^{\text{ALS}}. \end{aligned} \tag{1}$$

Output: The challenger forwards a bit $b' \in \{0, 1\}$ received from $\mathcal{A}$ to the AD-CPA challenger of the ALS.

By applying the leftover hash lemma (e.g. Theorem 8.38 of [54]), it is not hard to see that $\mathcal{A}$'s advantage in this hybrid is upper bounded by the advantage of the challenger of breaking the ALS scheme. Therefore, due to the AD-CPA security of the ALS scheme, we conclude that the adversary $\mathcal{A}$ has zero advantage in winning **Hybrid** 4.

Below we show that each two adjacent hybrids are indistinguishable.

Lemma 10. Hybrid 0 *and* **Hybrid** 1 *are statistically indistinguishable.*

Proof. The only difference between the two hybrids is in how the public matrix $\mathbf{U}$ is generated. In **Hybrid** 0, $\mathbf{U}$ is sampled uniformly at random from $\mathbb{Z}_q^{n \times m}$, that is, we have $\mathbf{U} \leftarrow \mathcal{U}$, where $\mathcal{U}$ is a uniform distribution over $\mathbb{Z}_q^{n \times m}$. Whereas, in **Hybrid** 1, we have $\mathbf{U} = \mathbf{AX}$ for $\mathbf{X} \leftarrow \mathcal{D}_{\mathbb{Z}^{m \times m}, \sqrt{\gamma^2 + s^2}}$. By Lemma 5 (item 2), we have $\mathbf{A} \overset{stat}{\approx} \mathcal{U}$ for $\mathbf{A}$ output by **TrapGen**. Followed by Lemma 2, we have $(\mathbf{A}, \mathbf{U}) \overset{stat}{\approx} (\mathbf{A}, \mathcal{U})$ for random matrix $\mathbf{A}$, which shows that the matrix $\mathbf{U}$ of **Hybrid** 0 is statistically indistinguishable from that in **Hybrid** 1. It follows that **Hybrid** 0 and **Hybrid** 1 are statistically indistinguishable.

Lemma 11. **Hybrid** 1 *and* **Hybrid** 2 *are statistically indistinguishable.*

Proof. The only difference between the two hybrids is in how the public matrices $(\mathbf{B}_1, \ldots, \mathbf{B}_\theta)$ are generated. In **Hybrid** 1, $(\mathbf{B}_1, \ldots, \mathbf{B}_\theta)$ are chosen uniformly at random from $\mathbb{Z}_q^{n \times m}$. Whereas, in **Hybrid** 2, we have $\mathbf{B}_i = \mathbf{A}\mathbf{S}_i^* - \mathsf{att}_i^* \mathbf{G}$ for $i \in [\theta]$. Moreover, note that in **Hybrid** 1, the challenger generates a challenge ciphertext $\mathbf{c}^*$ in the challenge phase, where $\{\mathbf{S}_i^*\}_{i \in [\theta]}$ are for the construction of $\mathbf{c}^*$; by contrast, the matrix $\mathbf{S}_i^*$ is used to generate $\mathbf{B}_i$ and the challenge ciphertext $\mathbf{c}^*$, where $\mathbf{e} := (\mathbf{I}_m \mid \mathbf{S}_1^* \mid \cdots \mid \mathbf{S}_\theta^*)^T \cdot \mathbf{e}_0$ is used as the error vector for some $\mathbf{e}_0 \in \mathcal{D}_{\mathbb{Z}^{2m}, \sigma}$.

By Lemma 3, we have $(\mathbf{A}, \{\mathbf{A}\mathbf{S}_i^*\}_{i \in [\theta]}, \mathbf{e}) \overset{stat}{\approx} (\mathbf{A}, \{\mathbf{A}_i\}_{i \in [\theta]}, \mathbf{e})$, where $\{\mathbf{A}_i\}_{i \in [\theta]} \leftarrow \mathbb{Z}_q^{n \times m}$, which shows that the public matrices $(\mathbf{B}_1, \ldots, \mathbf{B}_\theta)$ in **Hybrid** 1 and **Hybrid** 2 are statistically indistinguishable. It follows that **Hybrid** 1 and **Hybrid** 2 are statistically indistinguishable.

Lemma 12. **Hybrid** 2 *and* **Hybrid** 3 *are statistically indistinguishable.*

Proof. The only difference between the two hybrids is in how the public matrix $\mathbf{A}$ and the secret key for answering key queries are generated. First, in **Hybrid** 2, $\mathbf{A}$ is generated by $(\mathbf{A}, \mathbf{T}_\mathbf{A}) \leftarrow \mathbf{TrapGen}(1^n, m, q)$; while in **Hybrid** 3, $\mathbf{A}$ is sampled uniformly at random from $\mathbb{Z}_q^{n \times m}$. By Lemma 5 (item 1), $\mathbf{A}$ in **Hybrid** 2 and **Hybrid** 3 are statistically indistinguishable.

On the other hand, in **Hybrid** 2, the challenger uses the trapdoor $\mathbf{T}_\mathbf{A}$ to generate all secret keys for answering key queries, that is, for any pair $(f, \mathbf{y})$ submitted by $\mathcal{A}$, the challenger computes $\mathbf{B}_f = \mathbf{Eval}_{pk}(f, (\mathbf{B}_1, \ldots, \mathbf{B}_\theta))$, chooses $\mathbf{W} \leftarrow \mathcal{D}_{\mathbb{Z}^{m \times m}, \gamma}$, lets $\mathbf{D} = \mathbf{U} - \mathbf{A}\mathbf{W}$, and generates the secret key $sk_{f,\mathbf{y}} := (\mathbf{y}, \mathbf{R}_f \cdot \mathbf{y})$, where $\mathbf{R}_f = \begin{bmatrix} \mathbf{R}_1 + \mathbf{W} \\ \mathbf{R}_2 \end{bmatrix}$ and $\begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \end{bmatrix} \in \mathbb{Z}^{2m \times m} \leftarrow \mathbf{SampleLeft}(\mathbf{A}, \mathbf{B}_f, \mathbf{T}_\mathbf{A}, \mathbf{D}, s)$. Whereas, going back to **Hybrid** 3, the challenger generates the secret key $sk_{f,\mathbf{y}}$ by considering two cases (corresponding to two conditions that the key query must meet) without using the trapdoor $\mathbf{T}_\mathbf{A}$:

- In **Case** 1 of **Hybrid** 3, the challenger uses **SampleRight** algorithm with the public trapdoor $\mathbf{T}_\mathbf{G}$ to generate $\mathbf{R}_f$. Since $||\mathbf{S}_f^*||_2 < (m+1)^{d+1}$, it holds that $s \geq \sqrt{5} \cdot (||\mathbf{S}_f^*||_2 + 1) \cdot \omega(\sqrt{\log m})$ as required for **SampleRight** algorithm. Hence, by Lemma 5 (items 2 and 3), $\mathbf{R}_f$ of $sk_{f,\mathbf{y}}$ in **Hybrid** 2 and **Hybrid** 3 are statistically indistinguishable.
- In **Case** 2 of **Hybrid** 3, the challenger generates $\mathbf{R}_f = \begin{bmatrix} \mathbf{X} - \mathbf{S}_f^* \mathbf{R}_2 \\ \mathbf{R}_2 \end{bmatrix}$ for $\mathbf{R}_2 \leftarrow \mathcal{D}_{\mathbb{Z}^{m \times m}, s}$ using the fact that $\mathbf{U} = \mathbf{A}\mathbf{X}$ for $\mathbf{X} \leftarrow \mathcal{D}_{\mathbb{Z}^{m \times m}, \sqrt{\gamma^2 + s^2}}$. By rephrasing in the terms used in Lemma 6, the key generation in **Hybrid** 2 is exactly the procedure of **Sampler**-1, and in **Hybrid** 3 the **Sampler**-2. By Lemma 6, $\mathbf{R}_f$ of $sk_{f,\mathbf{y}}$ in **Hybrid** 2 and **Hybrid** 3 are statistically indistinguishable.

In conclusion, since the public matrices and responses to all key queries in **Hybrid** 2 are statistically indistinguishable from those in **Hybrid** 3, it follows that **Hybrid** 2 and **Hybrid** 3 are statistically indistinguishable.

Lemma 13. **Hybrid** 3 *and* **Hybrid** 4 *are statistically indistinguishable.*

Proof. First, since the public matrices $(\mathbf{A}, \mathbf{U})$ in **Hybrid** 4 are statistically close to uniform over $\mathbb{Z}_q^{n \times m}$ due to the AD-CPA security of the ALS scheme (or by Lemma 2), the public matrices in **Hybrid** 3 and **Hybrid** 4 are statistically indistinguishable. Consider the challenge ciphertext $(\mathbf{c}'_1, \mathbf{c}'_2)$ in **Hybrid** 3, which is of the form

$$\begin{aligned}\mathbf{c}'_1 &= \mathbf{H}^T\mathbf{s} + \mathbf{e}, \\ \mathbf{c}'_2 &= \mathbf{U}^T\mathbf{s} + \mathbf{e}_1 + \lfloor q/K \rfloor \cdot \mathbf{x}_b\end{aligned} \tag{2}$$

where

$$\begin{aligned}\mathbf{H} &= (\mathbf{A} \mid \mathbf{B}_1 + \mathsf{att}_1^*\mathbf{G} \mid \cdots \mid \mathbf{B}_\theta + \mathsf{att}_\theta^*\mathbf{G}) \\ &= (\mathbf{A} \mid \mathbf{AS}_1^* - \mathsf{att}_1^*\mathbf{G} + \mathsf{att}_1^*\mathbf{G} \mid \cdots \mid \mathbf{AS}_\theta^* - \mathsf{att}_\theta^*\mathbf{G} + \mathsf{att}_\theta^*\mathbf{G}) \\ &= (\mathbf{A} \mid \mathbf{AS}_1^* \mid \cdots \mid \mathbf{AS}_\theta^*).\end{aligned}$$

Then, we can transform Eq.(2) to Eq.(3)

$$\begin{aligned}\mathbf{c}'_1 &= (\mathbf{I}_m \mid \mathbf{S}_1^* \mid \cdots \mid \mathbf{S}_\theta^*)^T \cdot (\mathbf{A}^T\mathbf{s} + \mathbf{e}_0), \\ \mathbf{c}'_2 &= \mathbf{U}^T\mathbf{s} + \mathbf{e}_1 + \lfloor q/K \rfloor \cdot \mathbf{x}_b,\end{aligned} \tag{3}$$

where $\mathbf{e} = (\mathbf{I}_m \mid \mathbf{S}_1^* \mid \cdots \mid \mathbf{S}_\theta^*)^T \cdot \mathbf{e}_0$.

On the other hand, let $\mathbf{e}'_0, \mathbf{e}'_1$ be the error vectors of the ALS ciphertext $(\mathbf{c}_1^{\text{ALS}}, \mathbf{c}_2^{\text{ALS}})$, then we can rewrite Eq.(1) in **Hybrid** 4 as

$$\begin{aligned}\mathbf{c}_1 &= (\mathbf{I}_m \mid \mathbf{S}_1^* \mid \cdots \mid \mathbf{S}_\theta^*)^T \cdot (\mathbf{A}_{\text{ALS}}^T\mathbf{s} + \mathbf{e}'_0), \\ \mathbf{c}_2 &= \mathbf{U}_{\text{ALS}}^T\mathbf{s} + \mathbf{e}'_1 + \lfloor q/K \rfloor \cdot \mathbf{x}_b.\end{aligned} \tag{4}$$

Since $(\mathbf{A}, \mathbf{U}) \overset{stat}{\approx} (\mathbf{A}_{\text{ALS}}, \mathbf{U}_{\text{ALS}})$, we have that the ciphertext $(\mathbf{c}_1, \mathbf{c}_2)$ of Eq.(4) is statistically indistinguishable from the ciphertext $(\mathbf{c}'_1, \mathbf{c}'_2)$ of Eq.(3). In other words, the challenge ciphertext in **Hybrid** 3 and in **Hybrid** 4 are statistically indistinguishable. It follows that **Hybrid** 3 and **Hybrid** 4 are statistically indistinguishable.