Traitor Tracing with $N^{1/3}$-size Ciphertexts and $O(1)$-size Keys from k-Lin

Junqing Gong[1,2,⋆], Ji Luo[3,⋆⋆], and Hoeteck Wee[4]

[1] East China Normal University, Shanghai, China
jqgong@sei.ecnu.edu.cn
[2] Shanghai Qi Zhi Institute, Shanghai, China
[3] Paul G. Allen School of Computer Science & Engineering,
University of Washington, Seattle, USA
luoji@cs.washington.edu
[4] NTT Research, Sunnyvale, USA
wee@di.ens.fr

Abstract. We present a pairing-based traitor tracing scheme for N users with

$$|\mathsf{pk}| = |\mathsf{ct}| = O(N^{1/3}), \quad |\mathsf{sk}| = O(1).$$

This is the first pairing-based scheme to achieve $|\mathsf{pk}| \cdot |\mathsf{sk}| \cdot |\mathsf{ct}| = o(N)$. Our construction relies on the (bilateral) k-Lin assumption, and achieves private tracing and full collusion resistance. Our result simultaneously improves upon the sizes of pk, ct in Boneh–Sahai–Waters [Eurocrypt '06] and the size of sk in Zhandry [Crypto '20], while further eliminating the reliance on the generic group model in the latter work.

1 Introduction

Traitor tracing schemes [13] enable a content distributor to generate secret keys for different users, any of whom can decrypt some protected content (e.g., a cable TV stream). However, if any group of "traitors" get together and publish some program capable of decrypting the content, then it is possible to use this program to "trace" and identify at least one of the traitors and therefore hold them accountable. We would like to design traitor tracing schemes with short parameters, namely short public key pk, ciphertext overhead ct and secret key sk that depend minimally on the total number of users N.

In this work, we focus on pairing-based traitor tracing schemes, where the most efficient schemes achieve $|\mathsf{pk}| \cdot |\mathsf{sk}| \cdot |\mathsf{ct}| = \Theta(N)$, with the classic result of Boneh, Sahai, and Waters (BSW) [6] achieving $|\mathsf{pk}| = |\mathsf{ct}| = O(N^{1/2})$, $|\mathsf{sk}| = O(1)$ as well as a recent work by Zhandry [29] achieving $|\mathsf{pk}| = |\mathsf{ct}| = |\mathsf{sk}| = O(N^{1/3})$.

In view of the state of the art for pairing-based traitor tracing, Zhandry put forth the following conjecture, which captures the community's intuition about the optimal trade-offs for pairing-based traitor tracing:

> *For any $a, b, c \geq 0$ such that $a + b + c = 1$, there exists a pairing-based traitor tracing scheme with $|\mathsf{pk}| = O(N^a), |\mathsf{sk}| = O(N^b), |\mathsf{ct}| = O(N^c)$.*

He also proved the conjecture for (1) $b = 0, c \geq a$ and (2) $b, c \geq a$ [29]. In light of this conjecture, we consider two open problems in this work. The first is a special case of Zhandry's conjecture:

> *Does there exist a traitor tracing scheme with* $|\mathsf{pk}| = O(N^{2/3}), |\mathsf{ct}| = O(N^{1/3}), |\mathsf{sk}| = O(1)$ *from pairings?*

[⋆] Partially supported by National Natural Science Foundation of China (62002120), Innovation Program of Shanghai Municipal Education Commission (2021-01-07-00-08-E00101) and the "Digital Silk Road" Shanghai International Joint Lab of Trustworthy Intelligent Software (22510750100). Part of this work was done while visiting NTT Research.

[⋆⋆] Partially supported by NSF grants CNS-1936825 (CAREER), CNS-2026774, a JP Morgan AI Research Award, a Cisco Research Award, and a Simons Collaboration on the Theory of Algorithmic Fairness. Part of this work was done during an internship at NTT Research.

Such a scheme would inherit the short ciphertexts and short keys of the afore-mentioned pairing-based schemes in [6,29]. We note that minimizing secret key size is important in settings where the decryption devices have limited long-term cryptographic storage, and perhaps even more important than minimizing ciphertext overhead, since the total ciphertext size is often dominated by the size of the payload (megabytes) and not the ciphertext overhead coming from the traitor tracing scheme (kilobytes).

The next question challenges the optimality of Zhandry's conjecture:

Does there exist a traitor tracing scheme with $|\mathsf{pk}| \cdot |\mathsf{sk}| \cdot |\mathsf{ct}| = O(N^{1-\delta})$ *for some* $\delta > 0$ *from pairings?*

An affirmative answer would indicate that the trade-off suggested in Zhandry's conjecture is far from optimal, and more importantly, that our intuition about pairing-based traitor tracing is in fact flawed!

Traitor Tracing Beyond Pairings. Before we go on to describe our results, we note that from LWE (or obfuscation), we have "optimal" traitor tracing schemes achieving $|\mathsf{pk}| + |\mathsf{sk}| + |\mathsf{ct}| = \mathrm{poly}(\log N)$ [18,12,9]. Nonetheless, we believe there is still tremendous value in obtaining better pairing-based schemes. From a theoretical perspective, we want (i) good traitor tracing from different assumptions; (ii) to understand what's the best parameters we can get from pairings, as also considered in [29]; and (iii) to develop new tracing techniques, and indeed, the LWE-based schemes with $\mathrm{poly}(\log N)$ parameters rely crucially on ideas first developed in earlier pairing-based schemes. From a practical perspective, pairing-based cryptographic schemes are more widely deployed than lattice-based ones (e.g., in blockchain-type applications, and with better libraries, etc.) and for moderately small values of N that arise in applications, could potentially achieve better concrete efficiency than the asymptotically more efficient LWE-based schemes.

1.1 Our Results

We answer both open problems in the affirmative: we present a pairing-based traitor tracing scheme for N users with

$$|\mathsf{pk}| = |\mathsf{ct}| = O(N^{1/3}), \quad |\mathsf{sk}| = O(1).$$

This is the first pairing-based scheme to achieve $|\mathsf{pk}| \cdot |\mathsf{sk}| \cdot |\mathsf{ct}| = o(N)$. Our construction relies on the (bilateral) k-Lin assumption, and achieves private tracing and full collusion resistance. Our result simultaneously improve upon the sizes of pk, ct in [6] and the size of sk in [29], while further eliminating the reliance on the generic group model (GGM) in the latter work. As in Zhandry's work, the $O(\cdot)$ terms hides factors polynomial in the security parameter. See Fig. 1 for comparison with prior works.

1.2 Technical Overview

We proceed to provide a brief overview of our scheme and a technical comparison with Zhandry's construction [29]. In this overview, for any positive integer N, we define $[N] = \{1, 2, \ldots, N\}$ and $[0, N] = \{0, 1, \ldots, N\}$. Note that $[0] = \emptyset$.

Recap: PLBE and BSW Traitor Tracing. An N private linear broadcast encryption (N-PLBE) [6] is a type of anonymous broadcast encryption where we can revoke decryption capabilities for the first z users. In particular,

- key generation produces a key sk for each user identity $i \in [N]$;
- encryption takes as input a private index $z \in [0, N]$ and a message m to produce a ciphertext ct;
- decryption returns m if $i > z$, or equivalently, $i \notin [z]$.

The security requirements for PLBE are as follows:

- message-hiding: the message m is hidden given unauthorized keys;

Scheme	|pk|	|ct|	|sk|	**Assumption**	**Tracing**
folklore + IBE [29]	1	N	1	IBE	public
BN08 [5] *†	1	κ	$N^2\kappa^2$	IBE	private
BP08 [4] *	1	κ	$N^2\kappa$	IBE	private
BSW06 [6]	$\sqrt{N}$	$\sqrt{N}$	1	composite	private
BW06 [8]	$\sqrt{N}$	$\sqrt{N}$	$\sqrt{N}$	composite	public
PLBE + W20 [28]	$\sqrt{N}$	$\sqrt{N}$	1	bi-k-Lin	public
Z20 [29]	$\sqrt[3]{N\kappa^4}$	$\sqrt[3]{N\kappa^4}$	$\sqrt[3]{N\kappa^4}$	GGM	private
this work (§ 3)	$\sqrt[3]{N\kappa}$	$\sqrt[3]{N\kappa}$	κ	bi-k-Lin	private

Fig. 1. Comparison with prior pairing-based traitor tracing schemes for N users, where size L means $\Theta(L)$ group elements plus $O(L)$ bits. Here, κ denotes the statistical security parameter, with statistical error $2^{-\Omega(\kappa)}$. In the "**Assumption**" column, "bi-k-Lin" (bilateral k-Lin) is a strengthening of the k-Linear assumption in prime-order groups (equivalent to k-Linear for symmetric bilinear groups), "composite" stands for assumptions in composite-order symmetric bilinear groups (e.g., subgroup membership assumption), and "GGM" stands for generic group model.

* IBE is used to compress pk [29].

† Threshold elimination compiler [29] is applied.

- index-hiding: encryptions of $(z-1,m)$ and (z,m) are computationally indistinguishable given all secret keys for identities $i \neq z$.

Starting from an N-PLBE, BSW traitor tracing scheme [6] with identity space $[N]$ works as follows:

- The public key and secret keys are the same as for PLBE;
- An encryption of m is a PLBE encryption of m with $z = 0$.

Correctness is straight-forward: every secret key satisfies $i > 0$, or equivalently, $i \notin [0] = \emptyset$, and is authorized to recover m.

Given a decoder D with distinguishing advantage ε, we can identify a traitor $i^* \in [N]$ with probability negligibly close to 1 as follows: for $i = 0,1,\ldots,N$, revoke the decryption capabilities of the first i users by feeding the decoder PLBE encryption of (i,m). We know that the advantage of D is ε for $i = 0$ (by the fact that decoder is "good") and negligible for $i = N$ (by message-hiding). Therefore, there exists $i^* \in [N]$ such that there is a significant drop –at least roughly ε/N– in the distinguishing advantage of the decoder from i^*-1 to i^*; index-hiding ensures that i^* is a traitor.

The state of the art for N-PLBE achieves parameter sizes

$$|\mathsf{pk}| = O(N^{1/2}), \quad |\mathsf{ct}| = O(N^{1/2}), \quad |\mathsf{sk}| = O(1)$$

from the bilateral k-Lin, via functional encryption for quadratic functions [3,28]. The resulting traitor tracing scheme achieves the same parameter sizes.

Our Starting Point: Revocable PLBE. We will explain our traitor tracing scheme using a generalization of PLBE which we refer to as (N_1,N_2) revocable private linear broadcast encryption ((N_1,N_2)-rPLBE). In an (N_1,N_2)-rPLBE,

- key generation takes as input a user identity $(i_1,i_2) \in [N_1]\times[N_2]$ to produce a key sk;
- encryption takes as input a private index $z \in [0,N_1]$, a private set $S \subseteq [N_1]\times[N_2]$ of size at most N_2, and a message m to produce a ciphertext ct.
- decryption returns m if $(i_1,i_2) \notin ([z]\times[N_2]) \cup S$.

This allows us to revoke the decryption capability of the first $z \cdot N_2$ users as well as additional (at most) N_2 users in the set S. We call them *index-revocation* and *set-revocation*, respectively. Accordingly, the security requirements for rPLBE are generalized as follows:

- message-hiding: the message m is hidden given only unauthorized keys.
- index-hiding: encryptions of $(z-1, S, m)$ and (z, S, m) are indistinguishable, even given secret keys for all identities in $\{(i_1, i_2) \in [N_1] \times [N_2] : i_1 \neq z\} \cup S$.
- set-hiding: encryptions of (z, S_0, m) and (z, S_1, m) with $S_0 \subset S_1$ are indistinguishable, even given secret keys for all identities $(i_1, i_2) \notin S_1 \setminus S_0$.

Note that PLBE corresponds to the special case $N_2 = 1$ and $S = \emptyset$ during encryption. In this case, the identity is of the form $(i_1, 1)$; index-hiding reduces to that for PLBE; set-hiding becomes dummy since we always have $S_0 = S_1 = \emptyset$.

Tracing Using rPLBE. Starting from an (N_1, N_2)-rPLBE, we build a traitor tracing scheme with identity space $[N_1] \times [N_2]$ as follows:

- The public key and secret keys are the same as for rPLBE;
- An encryption of m is a rPLBE encryption of m with $z = 0, S = \emptyset$.

Correctness is straight-forward.

Given a decoder D with distinguishing advantage ε, our goal is to identify a traitor $(i_1^*, i_2^*) \in [N_1] \times [N_2]$ with probability negligibly close to 1. The tracing strategy proceeds in two steps:

Step 1: Identifying i_1^* via Index-Revocation. For $i_1 = 0, 1, \ldots, N_1$, we revoke the decryption capabilities of the first $i_1 \cdot N_2$ users by feeding the decoder rPLBE encryptions of $(i_1, \emptyset, m)$. As BSW traitor tracing with PLBE, there exists $i_1^* \in [N_1]$ such that there is a significant drop –at least roughly ε/N_1– in the distinguishing advantage of the decoder from i_1^*-1 to i_1^*, upon which we know that one of the users in $\{i_1^*\} \times [N_2]$ is a traitor by index-hiding security (applied to $z = i_1^*, S = \emptyset$).

Step 2: Identifying i_2^* via Set-Revocation. Next, for $i_2 = 0, \ldots, N_2$, we revoke the decryption capabilities of the first i_2 users in $\{i_1^*\} \times [N_2]$ by feeding the decoder rPLBE encryptions of $(i_1^* - 1, S_{i_1^*, i_2}, m)$ and $(i_1^*, S_{i_1^*, i_2}, m)$, where $S_{i_1^*, i_2}$ is $\{(i_1^*, j) : j \in [i_2]\}$, and define ε_{i_2} to be the difference between the distinguishing advantages for the two ciphertext distributions. We begin with the following bounds on ε_0 and ε_{N_2} (corresponding to $i_2 = 0$ and $i_2 = N_2$ respectively):

(1) $\varepsilon_0 \gtrsim \varepsilon/N_1$. This follows from Step 1 and the fact that $S_{i_1^*,0} = \emptyset$.

(2) ε_{N_2} is negligible. This follows from applying index-hiding to $z = i_1^*$ and $S = S_{i_1^*, N_2}$, and holds even when the adversary gets the secret keys for all possible identities since $\{(i_1, i_2) \in [N_1] \times [N_2] : i_1 \neq i_1^*\} \cup S_{i_1^*, N_2} = [N_1] \times [N_2]$.

Therefore, there exists $i_2^* \in [N_2]$ such that $\varepsilon_{i_2^*} - \varepsilon_{i_2^*-1} \gtrsim \varepsilon/N_1 N_2$. By set-hiding applied to $z = i_1^* - 1, S_0 = S_{i_1^*, i_2^*-1}$, $S_1 = S_{i_1^*, i_2^*}$ and $z = i_1^*, S_0 = S_{i_1^*, i_2^*-1}, S_1 = S_{i_1^*, i_2^*}$, the user (i_1^*, i_2^*) must be a traitor. Note that, for Step 2, we always have $|S| \leq N_2$.

Implementing Set-Revocation and Set-Hiding. For set-revocation, we will need to relax the syntax for (N_1, N_2)-rPLBE as follows (following "mixed functional encryption" in [18]) :

- encrypting to arbitrary sets S requires knowledge of msk;
- encrypting to $S = \emptyset$ only requires mpk.

As a result of this relaxation, our traitor tracing scheme only achieves private tracing, as is also the case in [18] and in Zhandry's work [29].

One-Ciphertext Security. Consider the following construction for set-revocation (i.e., ignoring the index-revocation) for N_2 users (cf. Step 2 of our tracing) based on any negated-IBE scheme where a key for $\mathsf{id} \in \{0,1\}^\kappa$ can decrypt a ciphertext for $\mathsf{id}' \in \{0,1\}^\kappa$ iff $\mathsf{id} \neq \mathsf{id}'$.

- The public key consists of N_2 independent public keys for negated-IBE $\mathsf{mpk}_1, \ldots, \mathsf{mpk}_{N_2}$.

- The secret key for user i_2 is a random $u_{i_2} \leftarrow \{0,1\}^\kappa$, together with a negated-IBE key for u_{i_2} w.r.t. mpk_{i_2}. This means that u_{i_2} is perfectly hidden from the decoder if user i_2 is honest.
- To encrypt to $S \subseteq [N_2]$, we provide N_2 negated-IBE ciphertexts for identities $r_1, \dots, r_{N_2}$ w.r.t. $\mathsf{mpk}_1, \dots, \mathsf{mpk}_{N_2}$ respectively, where $r_{i_2} \leftarrow \{0,1\}^\kappa$ if $i_2 \notin S$, and $r_{i_2} = u_{i_2}$ if $i_2 \in S$.

Correctness is straight-forward: $\Pr[r_{i_2} \neq u_{i_2} : r_{i_2} \leftarrow \{0,1\}^\kappa] = 1 - 2^{-\kappa}$. Set-hiding for a single ciphertext follows from the fact that (1) if the adversary does not see the key for user i_2, then u_{i_2} is statistically random and (2) we always have N_2 negated-IBE ciphertexts (which hides $|S|$).

Multi-Ciphertext Security via Threshold Broadcast. In order to achieve set-hiding for multiple ciphertexts, as is necessary for tracing, we adopt Zhandry's "threshold broadcast" technique. We will rely on an *approximated* version of negated-IBE where $\mathsf{id} \neq \mathsf{id}'$ (i.e., $\mathsf{wt}(\mathsf{id} \oplus \mathsf{id}') > 0$) is replaced with $\mathsf{wt}(\mathsf{id} \oplus \mathsf{id}') \geq 2\kappa/5$ where $\mathsf{wt}(\cdot)$ corresponds to Hamming weight. To revoke a user $i_2 \in S$ while preserving set-hiding, we will sample r_i from a carefully-designed distribution of bit-strings close to u_i in Hamming distance, where the distribution depends adaptively on the adversary (c.f. Lemma 1 in Section 3.2); for this reason, we will require adaptive security w.r.t. id.

Instantiating rPLBE: Warm-Up. Next, we translate the problem of building an (N_1, N_2)-rPLBE to a problem about functional encryption, specifically, that of attribute-based functional encryption (AB-FE) [1]. In AB-FE, a ciphertext is associated with a private attribute z and a public attribute x, a key with a function f and a predicate P, and decryption returns $f(z)$ if $P(x)$ is true. Specifically, an (N_1, N_2)-rPLBE implementing threshold-broadcast-based set-revocation (sketched above) would follow from AB-FE for

$$f_{i_1}^{\text{comp}}(\overbrace{j_1, m}^{z}) = \begin{cases} m, & \text{if } i_1 > j_1; \\ 0, & \text{otherwise.} \end{cases}$$

$$P_{i_2,u}^{\text{tbe}}(\overbrace{r_1, \dots, r_{N_2}}^{x}) = \begin{cases} 1, & \text{if } \mathsf{wt}(r_{i_2} \oplus u) \geq 2\kappa/5; \\ 0, & \text{otherwise;} \end{cases}$$

where $f_{i_1}^{\text{comp}}$ implements index-revocation and index-hiding, and $P_{i_2,u}^{\text{tbe}}$, set-revocation. Recall that set-hiding relies on the distribution of $r_1, \dots, r_{N_2}$.

The recent work of Abdalla, Catalano, Gay, and Ursu (ACGU) [1] presented an AB-FE scheme based on the k-Lin assumption for the setting where f corresponds to inner product and P corresponds to read-once span programs. It is easy to see that we can implement $f_{i_1}^{\text{comp}}$ as an inner product over vectors of length $O(N_1)$, and $P_{i_2,u}^{\text{tbe}}$ as a read-once span program of size $O(N_2\kappa)$. Combined with the ACGU result, we obtain an (N_1, N_2)-rPLBE with

$$|\mathsf{pk}| = O(N_1 + N_2\kappa), \quad |\mathsf{ct}| = O(N_1 + N_2\kappa), \quad |\mathsf{sk}| = O(N_2\kappa)$$

The parameter sizes are essentially the sum of those for (i) inner product FE for vectors of length $\ell = N_1$, namely $|\mathsf{pk}| = |\mathsf{ct}| = O(\ell), |\mathsf{sk}| = O(1)$, and (ii) ABE for read-once span programs of size $s = N_2\kappa$, namely $|\mathsf{pk}| = |\mathsf{ct}| = |\mathsf{sk}| = O(s)$.

Instantiating rPLBE: Ours. We present an (N_1, N_2)-rPLBE based on (bilateral) k-Lin achieving shorter parameters

$$|\mathsf{pk}| = O(\underline{N_1^{1/2}} + N_2\kappa), \quad |\mathsf{ct}| = O(\underline{N_1^{1/2}} + N_2\kappa), \quad |\mathsf{sk}| = O(\underline{\kappa}),$$

we highlight the improvements by underlines. Setting $N_1 = N^{2/3}, N_2 = N^{1/3}$ yields our main result. We achieve shorter parameters as follows:

- To reduce the dependency on N_1 in pk, ct to $N_1^{1/2}$, we implement $f_{i_1}^{\text{comp}}$ using quadratic functions over inputs of length $N_1^{1/2}$, following [6,3].
- To reduce $|\mathsf{sk}|$ to $O(\kappa)$, we observe that the span program computing $P_{i_2,u}^{\text{tbe}}$ is κ-local, that is, it depends only on κ bits of its input and show that for such span programs, the ABE key size can be decreased to $O(\kappa)$.

To put these two pieces together, we combine the ACGU construction which only supports linear functions over the private attribute z with techniques from functional encryption for quadratic functions [28].

Achieving Adaptive Security from k-Lin. We need an additional idea to achieve adaptive security w.r.t. r's, which is necessary for our traitor tracing strategy. The challenge lies in the fact that current techniques for realizing ABE adaptive security from the k-Lin assumption via dual system encryption methodology [26] rely on the guarantee (provided by the ABE security game) that the predicate P is never satisfied to switch the secret key distribution in the security proof. In the AB-FE security game, this guarantee goes away. To address this challenge, we observe that it suffices to construct AB-FE secure under a selective choice of z and an adaptive choice of x, since z (ignoring the payload m) comes from a polynomial-size domain. When a key query for f, P comes along, we will decide whether to switch the secret key distribution depending on $f(z)$. More precisely, the security experiment requires that an adversary selectively specifies z_0, z_1, we only switch the secret key distribution (to a "semi-functional" key) if $f(z_0) \neq f(z_1)$, for which $P(x)$ must be false.

Comparison with Zhandry's $O(N^{1/3})$ Scheme. We provide a simplified overview of Zhandry's traitor tracing scheme [29]. The first step is a traitor tracing scheme for $N_1 N_2$ users with parameters:

$$|\mathsf{pk}| = O(N_2), \quad |\mathsf{ct}| = O(N_2\kappa), \quad |\mathsf{sk}| = O(N_1 + N_2\kappa)$$

The underlying scheme is a variant of an (N_1, N_2)-rPLBE, which is adaptively secure in the generic group model.[5]

Observe that the total parameter size for this scheme is $O(N_1 + N_2)$, similar to that based on ACGU, whereas we achieve total parameter size $O(N_1^{1/2} + N_2)$. The construction uses ideas from mixed bit matching encryption [17] (MBME), which can be instantiated from inner product predicate encryption.[6] In contrast, we crucially rely on techniques from quadratic FE to achieve the square-root dependency on N_1.

The second step in [29] is to amplify this to a traitor tracing scheme for $N_1 N_2 N_3$ users with parameters:

$$|\mathsf{pk}| = O(N_2), \quad |\mathsf{ct}| = O(N_2\kappa + N_3), \quad |\mathsf{sk}| = O(N_1 + N_2\kappa)$$

Setting $N_1 = N_2 = N_3 = N^{1/3}$ yields a traitor tracing scheme for N users with $|\mathsf{pk}| = |\mathsf{ct}| = |\mathsf{sk}| = O(N^{1/3})$.

Note that in addition to achieving better parameters and assumptions, our approach also streamlines Zhandry's approach, eliminating the use of MBME and risky tracing [17] and the second step above. Moreover, our scheme supports partially public tracing, in the sense that we can publicly identify a prefix $i_1^* \in [N_1]$ specifying a subset of $N_2 = N^{1/3}$ identities, one of which must be a traitor (via PLBE). This could be useful in applications where identity prefixes constitute important information, like country of origin or name of company.

Nonetheless, we stress that our results do not completely subsume those in [29]. In particular, the latter achieves some parameter trade-offs that we do not immediately achieve using our techniques, for instance, $|\mathsf{pk}| = O(1)$, $|\mathsf{sk}| = O(N), |\mathsf{ct}| = O(1)$ or $|\mathsf{pk}| = O(N^{1/4}), |\mathsf{sk}| = O(1), |\mathsf{ct}| = O(N^{3/4})$. Also, we do not present any broadcast-and-trace schemes.

[5] In a bit more detail, the construction starts with a variant of $(O(1), N_2)$-rPLBE with parameters

$$|\mathsf{pk}| = O(N_2), \quad |\mathsf{ct}| = O(N_2\kappa), \quad |\mathsf{sk}| = O(1)$$

which yields a "$1/N_1$-risky" traitor tracing scheme for $N_1 N_2$ users following [17]. That is, tracing succeeds with probability $1/N_1$. This is then amplified to a standard traitor tracing scheme with a blow-up in sk.

[6] In MBME, ciphertexts are associated with $(z_1, \ldots, z_\ell) \in \{0,1\}^\ell$ and keys with $(y_1, \ldots, y_\ell) \in \{0,1\}^\ell$ and decryption is possible iff

$$\wedge_{i=1}^{\ell} z_i \vee y_i = 1$$

Security requires both attribute and function hiding. MBME for ℓ-bit vectors can be instantiated from attribute-hiding function-hiding inner product predicate encryption for $O(\ell)$-dimensional vectors, since

$$\wedge_{i=1}^{\ell} z_i \vee y_i = 1 \Longleftrightarrow \sum_{i=1}^{\ell} (1 - z_i)(1 - y_i) \overset{?}{=} 0$$

1.3 Discussion

Open Problems. We conclude with several open problems:

- Combined with Zhandry's conjecture which asserts that we should be able to achieve full range of parameters with the same $|\mathsf{pk}| \cdot |\mathsf{sk}| \cdot |\mathsf{ct}|$, our result raises the tantalizing possibility of a pairing-based traitor tracing scheme with total parameter size $O(N^{2/9})$. In fact, it seems entirely plausible to have a pairing-based traitor tracing scheme with total parameter size $O(N^{1/4})$.
- Can we extend our techniques to broadcast with tracing following [19]? Or to public tracing with smaller parameters than in [6]? For public tracing from pairings, we conjecture that BSW is essentially optimal, namely we need $\min(|\mathsf{ct}|, |\mathsf{pk}| \cdot |\mathsf{sk}|) = \Omega(\sqrt{N})$.

Organization. We provide preliminaries in Section 2. Our traitor tracing based on AB-FE is given out in Section 3. We develop the AB-FE scheme required by the traitor tracing in Section 4.

2 Preliminaries

Notations. We denote by $s \leftarrow S$ the fact that s is picked uniformly at random from a finite set S. We use $\approx_s$ to denote two distributions being statistically indistinguishable, and $\approx_c$ to denote two distributions being computationally indistinguishable. We use lower-case boldfaced letters to denote *row* vectors and upper-case boldfaced letters to denote matrices. We use $\mathbf{e}_i$ to denote the i^{th} elementary row vector (with 1 at the i'th position and 0 elsewhere, and the total length of the vector specified by the context). For any positive integer N, we use $[N]$ to denote $\{1, 2, \ldots, N\}$ and $[0, N]$ to denote $\{0, 1, \ldots, N\}$.

2.1 Prime-Order Bilinear Groups

A group generation algorithm $\mathcal{G}$ takes as input the security parameter 1^λ and outputs a description $\mathbb{G} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$, where p is a prime, $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ are cyclic groups of order p, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a non-degenerate bilinear map. We require that the group operations in $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ and the bilinear map e be computable in deterministic polynomial time in λ. Let $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$, and $g_T = e(g_1, g_2) \in \mathbb{G}_T$ be the respective generators. We employ the *implicit representation* of group elements: for a matrix $\mathbf{M}$ over $\mathbb{Z}_p$, we define $[\mathbf{M}]_1 := g_1^{\mathbf{M}}, [\mathbf{M}]_2 := g_2^{\mathbf{M}}, [\mathbf{M}]_T := g_T^{\mathbf{M}}$, where exponentiation is carried out component-wise. Also, given $[\mathbf{A}]_1, [\mathbf{B}]_2$, we let $e([\mathbf{A}]_1, [\mathbf{B}]_2) = [\mathbf{AB}]_T$. We recall the matrix Diffie–Hellman (MDDH) assumption in $\mathbb{G}_1$ [14]:

Assumption 1 (MDDH$^d_{k,\ell}$) *Let $k, \ell, d \in \mathbb{N}$. We say that the* $\mathrm{MDDH}^d_{k,\ell}$ *assumption holds in $\mathbb{G}_1$ if for all p.p.t. adversary $\mathcal{A}$,*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathrm{MDDH}^d_{k,\ell}}(\lambda) := \Big| \Pr[\mathcal{A}(1^\lambda, \mathbb{G}, [\mathbf{A}]_1, \boxed{[\mathbf{SA}]_1}) = 1] - \Pr[\mathcal{A}(1^\lambda, \mathbb{G}, [\mathbf{A}]_1, [\mathbf{C}]_1) = 1] \Big|$$

is negligible in λ, where $\mathbb{G} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda)$, $\mathbf{A} \leftarrow \mathbb{Z}_p^{k \times \ell}$, $\mathbf{S} \leftarrow \mathbb{Z}_p^{d \times k}$, $\mathbf{C} \leftarrow \mathbb{Z}_p^{d \times \ell}$.

The MDDH assumption in $\mathbb{G}_2$ can be defined analogously. Escala *et al.* [14] showed that

$$k\text{-Lin} \Rightarrow \mathrm{MDDH}^1_{k,k+1} \Rightarrow \mathrm{MDDH}^d_{k,\ell} \quad \forall k, d \geq 1$$

When $\ell \leq k$, the $\mathrm{MDDH}^d_{k,\ell}$ assumption holds unconditionally.

Assumption 2 (bilateral MDDH$^d_{k,\ell}$) *Let $k, \ell, d \in \mathbb{N}$. We say that the bilateral* $\mathrm{MDDH}^d_{k,\ell}$ *assumption holds in $\mathbb{G}_1, \mathbb{G}_2$ if for all p.p.t. adversary $\mathcal{A}$,*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathrm{biMDDH}^d_{k,\ell}}(\lambda) := \Big| \Pr[\mathcal{A}(1^\lambda, \mathbb{G}, [\mathbf{A}]_1, \boxed{[\mathbf{SA}]_1}, [\mathbf{A}]_2, \boxed{[\mathbf{SA}]_2}) = 1] - \Pr[\mathcal{A}(1^\lambda, \mathbb{G}, [\mathbf{A}]_1, [\mathbf{C}]_1, [\mathbf{A}]_2, [\mathbf{C}]_2) = 1] \Big|$$

is negligible in λ, where $\mathbb{G} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda)$, $\mathbf{A} \leftarrow \mathbb{Z}_p^{k \times \ell}$, $\mathbf{S} \leftarrow \mathbb{Z}_p^{d \times k}$, $\mathbf{C} \leftarrow \mathbb{Z}_p^{d \times \ell}$.

The bilateral MDDH assumption is a strengthening of the MDDH assumption for asymmetric bilinear groups. It cannot hold for $k = 1$ for reasons similar to why DDH cannot hold in symmetric bilinear groups. An implication similar to that due to Escala *et al.* [14] holds:

$$\text{bilateral } k\text{-Lin} \Rightarrow \text{bilateral MDDH}^1_{k,k+1} \Rightarrow \text{bilateral MDDH}^d_{k,\ell} \quad \forall k \geq 2, d \geq 1.$$

By the implication, we will work with (bilateral) $\text{MDDH}^1_{k,k+1}$. This is sufficient for deriving our results based on (bilateral) k-Lin.

2.2 Traitor Tracing

We follow the definition in [29]. A traitor tracing scheme with key space $\mathcal{K}$ consists of four p.p.t. algorithms:

- $\mathsf{Gen}(1^\lambda, 1^N) \to \big(\mathsf{pk}, \mathsf{tk}, \{\mathsf{sk}_i\}_{i\in[N]}\big)$: The key generation algorithm takes the security parameter 1^λ and the number 1^N of users as input. It outputs a public key pk, a tracing key tk, and secret keys $\{\mathsf{sk}_i\}_{i\in[N]}$ (one for each user).
- $\mathsf{Enc}(\mathsf{pk}) \to (\mathsf{ct}, k)$: The encapsulation algorithm takes pk as input and outputs a ciphertext ct and an encapsulated key $k \in \mathcal{K}$.
- $\mathsf{Dec}(\mathsf{pk}, \mathsf{sk}_i, \mathsf{ct}) \to k$: The decapsulation algorithm takes $\mathsf{pk}, \mathsf{sk}_i, \mathsf{ct}$ as input and outputs a decapsulated key k.
- $\mathsf{Trace}^D(\mathsf{pk}, \mathsf{tk}, 1^{1/\varepsilon}) \to i^*$: The tracing algorithm takes pk, tk, and the error parameter $1^{1/\varepsilon}$ as input. It has oracle access to a decoder D and outputs a traitor identity $i^* \in [N]$ or $\perp$.

Correctness. We require that for all $c \in \mathbb{N}$, there exists a negligible function $\varepsilon(\lambda)$ such that for all $\lambda \in \mathbb{N}$, $N \in [\lambda^c]$, $i \in [N]$,

$$\Pr\left[\begin{array}{c}\big(\mathsf{pk}, \mathsf{tk}, \{\mathsf{sk}_i\}_{i\in[N]}\big) \leftarrow \mathsf{Gen}(1^\lambda, 1^N) \\ (\mathsf{ct}, k) \leftarrow \mathsf{Enc}(\mathsf{pk})\end{array} : \mathsf{Dec}(\mathsf{pk}, \mathsf{sk}_i, \mathsf{ct}) = k\right] \geq 1 - \varepsilon(\lambda).$$

Tracing Security. The scheme is secure if for all $\varepsilon(\lambda) > 0$ such that $1/\varepsilon(\lambda)$ is polynomially bounded, all efficient adversary $\mathcal{A}$ wins the following game with negligible probability:

- Launch $\mathcal{A}(1^\lambda)$ and receive 1^N from it. Run $\big(\mathsf{pk}, \mathsf{tk}, \{\mathsf{sk}_i\}_{i\in[N]}\big) \leftarrow \mathsf{Gen}(1^\lambda, 1^N)$ and send pk to $\mathcal{A}$.
- $\mathcal{A}$ adaptively queries keys for $i_q \in [N]$. Upon this query, send sk_{i_q} to $\mathcal{A}$. This stage can be repeated as many times as $\mathcal{A}$ wants. Let T be the set of i_q's for which the key is queried.
- $\mathcal{A}$ outputs a decoder D. Run $i^* \leftarrow \mathsf{Trace}^D(\mathsf{pk}, \mathsf{tk}, 1^{1/\varepsilon(\lambda)})$. $\mathcal{A}$ wins if

  $$\Pr\Big[b \leftarrow \{0,1\},\ k_0 \leftarrow \mathcal{K},\ (\mathsf{ct}, k_1) \leftarrow \mathsf{Enc}(\mathsf{pk}) : D^*(\mathsf{ct}, k_b) = b\Big] - \frac{1}{2} \geq \varepsilon(\lambda)$$

  and $i^* = \perp$, or if $i^* \notin T \cup \{\perp\}$.

Note that tracing security implies standard semantic security, cf. [29, Remark 3].

2.3 Attribute-Based Functional Encryption

An attribute-based functional encryption (AB-FE) for

$$\text{function class } \mathcal{F} = \{f : \mathcal{Z} \to \{0,1\}^*\} \text{ and predicate } P : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$$

consists of four p.p.t. algorithms:

- $\mathsf{Setup}(1^\lambda, \mathcal{F}, \mathcal{Z}, \mathcal{X}, \mathcal{Y}) \to (\mathsf{mpk}, \mathsf{msk})$: The set-up algorithm takes the security parameter 1^λ and the domains $\mathcal{F}, \mathcal{Z}, \mathcal{X}, \mathcal{Y}$ as input, and outputs a master public/secret key pair $(\mathsf{mpk}, \mathsf{msk})$.
- $\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, f, y) \to \mathsf{sk}$: The key generation algorithm takes mpk, msk, $f \in \mathcal{F}$, $y \in \mathcal{Y}$ as input, and outputs a secret key sk.
- $\mathsf{Enc}(\mathsf{mpk}, z, x) \to \mathsf{ct}$: The encryption algorithm takes mpk, $z \in \mathcal{Z}$, $x \in \mathcal{X}$ as input, and outputs a ciphertext ct.
- $\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}, f, y, \mathsf{ct}, x) \to d$: The decryption algorithm takes $\mathsf{mpk}, \mathsf{sk}, f, y, \mathsf{ct}, x$ as input and outputs $d \in \{0,1\}^*$.

Correctness. For all $\lambda \in \mathbb{N}$, $\mathcal{F}$, $\mathcal{Z}$, $\mathcal{X}$, $\mathcal{Y}$, $f \in \mathcal{F}$, $y \in \mathcal{Y}$, $z \in \mathcal{Z}$, $x \in \mathcal{X}$ such that $P(x,y) = 1$, we require

$$\Pr\left[\begin{array}{r}(\mathsf{mpk},\mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda,\mathcal{F},\mathcal{Z},\mathcal{X},\mathcal{Y})\\ \mathsf{sk} \leftarrow \mathsf{KeyGen}(\mathsf{mpk},\mathsf{msk},f,y) : \mathsf{Dec}(\mathsf{mpk},\mathsf{sk},f,y,\mathsf{ct},x) = f(z)\\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk},z,x) \qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\end{array}\right] = 1.$$

Our scheme will be based on pairing, for which we require f to take values in $\mathbb{Z}_p$ and relax the correctness requirement so that Dec only needs to output $[f(z)]_\mathrm{T}$.

Indistinguishability Security with Adaptive *x* and Semi-adaptive *z*. For all p.p.t. stateful $\mathcal{A}$, we require

$$\Pr\left[\begin{array}{c} b \leftarrow \{0,1\}\\ (\mathcal{F},\mathcal{Z},\mathcal{X},\mathcal{Y}) \leftarrow \mathcal{A}(1^\lambda)\\ (\mathsf{mpk},\mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda,\mathcal{F},\mathcal{Z},\mathcal{X},\mathcal{Y})\\ (z_0,z_1) \leftarrow \mathcal{A}(\mathsf{mpk})\\ x \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{mpk},\mathsf{msk},\cdot,\cdot)}()\\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk},z_b,x)\\ : \quad \mathcal{A}^{\mathsf{KeyGen}(\mathsf{mpk},\mathsf{msk},\cdot,\cdot)}(\mathsf{ct}) = b \end{array}\right] - \frac{1}{2}$$

to be negligible, where for each query (f_q, y_q) made to KeyGen by $\mathcal{A}$, it is required that $f_q(z_0) = f_q(z_1)$ or $P(x, y_q) = 0$.

We also consider a strengthened notion with partially adaptive z, where part of z_0, z_1 can be chosen with x, after querying arbitrarily many keys.

3 Building Traitor Tracing

We define threshold broadcast, private linear broadcast encryption (TB-PLBE), a certain kind of AB-FE, and use it to construct our traitor tracing scheme. We provide a construction for AB-FE that can be instantiated to TB-PLBE in Section 4; the instantiation can be found in Section 4.5.

3.1 TB-PLBE

We define TB-PLBE as an AB-FE (Section 2.3) for the function class

$$\mathcal{Z} = [0,N_1] \times \mathbb{Z}_p, \qquad \mathcal{F}^{\mathrm{comp}}_{N_1} = \left\{ f^{\mathrm{comp}}_{i_1} : [0,N_1] \times \mathbb{Z}_p \to \mathbb{Z}_p \;\middle|\; i_1 \in [N_1] \right\}$$

$$\forall i_1 \in [N_1], \quad f^{\mathrm{comp}}_{i_1}(j_1, m) = \begin{cases} m, & \text{if } i_1 > j_1;\\ 0, & \text{otherwise.}\end{cases}$$

and the predicate

$$\mathcal{X} = (\{0,1\}^\kappa)^{N_2}, \qquad \mathcal{Y} = [N_2] \times \{0,1\}^\kappa,$$

$$P^{\mathrm{tbe}}_{N_2,\kappa}\big((r_1,\ldots,r_{N_2}),(i_2,u)\big) = \begin{cases} 1, & \text{if } \mathrm{wt}(r_{i_2} \oplus u) \ge 2\kappa/5;\\ 0, & \text{otherwise;}\end{cases}$$

In Setup, the functionality $(\mathcal{F}^{\mathrm{comp}}_{N_1}, \mathcal{Z}, \mathcal{X}, \mathcal{Y})$ is represented by $(1^{N_1}, 1^{N_2})$. In KeyGen and Dec, the function $f^{\mathrm{comp}}_{i_1}$ is represented by i_1. We need a TB-PLBE secure under adaptively chosen $j_1, r_1, \ldots, r_{N_2}$ and semi-adaptively chosen m (cf. Section 2.3). We will present a construction of TB-PLBE in Section 4 that is secure under adaptively chosen $r_1, \ldots, r_{N_2}$ and selectively chosen j_1, m based on bi-k-Lin; the fact that $N_1 = \mathrm{poly}(\lambda)$ implies security against adaptively chosen j_1 by a standard guessing argument.

Remark 1 (relation with rPLBE). We will build traitor tracing from TB-PLBE directly in Section 3.2. For completeness, we briefly sketch how to implement (N_1, N_2)-rPLBE (with set-revocation suitable for our tracing algorithm) as outlined in Section 1.2 from TB-PLBE which is a AB-FE for $\mathcal{F}^{\text{comp}}_{N_1}$ and $P^{\text{tbe}}_{N_2,\kappa}$:

- The public key are the same as TB-PLBE; the secret key for user $(i_1, i_2) \in [N_1] \times [N_2]$ consists of a TB-PLBE key for $(i_1; i_2, u_{i_1,i_2})$ where u_{i_1,i_2} is fresh for each user.
- An encryption of (z, S, m) where $S \subseteq \{i_1^*\} \times [N_2]$ is a TB-PLBE encryption of $(z; (r_1, \ldots, r_{N_2}), m)$ where we sample r_{i_2} uniformly for $(i_1^*, i_2) \notin S$ but sample r_{i_2} according to the distribution $\rho_{i_1^*,i_2}$ for $(i_1^*, i_2) \in S$ described in Section 3.2.

Revocation mechanisms are as follows:

- The index-revocation for z relies on the function $\mathcal{F}^{\text{comp}}_{N_1}$ of TB-PLBE: $i_1 > z$ iff $(i_1, i_2) \notin [z] \times [N_2]$; index-hiding follows from the security of TB-PLBE, namely z is hidden.
- The set-revocation S relies on the predicate $P^{\text{tbe}}_{N_2,\kappa}$ and property of the distribution ρ_{i_1,i_2}: this ensures that $\mathsf{wt}(r_{i_2} \oplus u_{i_1^*,i_2}) < 2\kappa/5$ for all $(i_1^*, i_2) \in S$. However, set-hiding require an additional property of those distributions: distributions ρ_{i_1,i_2} are quite close to random distribution without the knowledge of u_{i_1,i_2}. See Lemma 1 for the two properties of ρ_{i_1,i_2}.

Note that we need the knowledge of u_{i_1,i_2} for finding ρ_{i_1,i_2} (see algorithm Learn in Lemma 1), therefore the encryption need secret key when $S \neq \emptyset$ for tracing (Step 2).

3.2 Traitor Tracing from TB-PLBE

Traitor Tracing Scheme. Let TBPLBE be a TB-PLBE scheme as defined in Section 3.1. Our traitor tracing scheme works as follows:

- $\mathsf{Gen}(1^\lambda, 1^N)$ sets κ to be any $\omega(\log\lambda)$ function that is polynomially bounded by λ, and $N_1 = N^{2/3}\kappa^{2/3}$, $N_2 = N^{1/3}\kappa^{-1/3}$. Treating the identity space $[N]$ as $[N_1] \times [N_2]$, the algorithm samples $u_{i_1,i_2} \leftarrow \{0,1\}^\kappa$ for all $i_1 \in [N_1]$ and $i_2 \in [N_2]$, runs

$$\begin{aligned}(\mathsf{tpmpk}, \mathsf{tpmsk}) &\leftarrow \mathsf{TBPLBE.Setup}(1^\lambda, 1^{N_1}, 1^{N_2}),\\ \mathsf{tpsk}_{i_1,i_2} &\leftarrow \mathsf{TBPLBE.KeyGen}(\mathsf{tpmsk}, i_1, (i_2, u_{i_1,i_2})),\\ \mathsf{sk}_{i_1,i_2} &\leftarrow (\mathsf{tpsk}_{i_1,i_2}, i_1, (i_2, u_{i_1,i_2})), \qquad \forall i_1 \in [N_1], i_2 \in [N_2],\end{aligned}$$

and outputs

$$\mathsf{pk} = \mathsf{tpmpk}, \quad \mathsf{tk} = \{u_{i_1,i_2}\}_{i_1 \in [N_1], i_2 \in [N_2]}, \quad \{\mathsf{sk}_{i_1,i_2}\}_{i_1 \in [N_1], i_2 \in [N_2]}.$$

- $\mathsf{Enc}(\mathsf{pk})$ samples $m \leftarrow \mathbb{Z}_p$ and $r_{j_2} \leftarrow \{0,1\}^\kappa$ for $j_2 \in [N_2]$. It runs

$$\mathsf{tpct} \leftarrow \mathsf{TBPLBE.Enc}(\mathsf{tpmpk}, (0, m), (r_1, \ldots, r_{N_2}))$$

and outputs $\mathsf{ct} = (\mathsf{tpct}, r_1, \ldots, r_{N_2})$ and $k = [m]_{\mathrm{T}}$. (Here, j_1 is set to 0.)
- $\mathsf{Dec}(\mathsf{sk}_{i_1,i_2}, \mathsf{ct})$ first parses sk_{i_1,i_2} into $(\mathsf{tpsk}_{i_1,i_2}, i_1, (i_2, u_{i_1,i_2}))$ and ct into $(\mathsf{tpct}, r_1, \ldots, r_{N_2})$. It outputs

$$\mathsf{TBPLBE.Dec}(\mathsf{tpsk}_{i_1,i_2}, i_1, (i_2, u_{i_1,i_2}), \mathsf{tpct}, (r_1, \ldots, r_{N_2})).$$

- $\mathsf{Trace}^D(\mathsf{pk}, \mathsf{tk}, 1^{1/\varepsilon})$ is described later.

Correctness. The correctness follows from that of TB-PLBE scheme with the fact that (i) $0 = j_1 < i_1$ for all $i_1 \in [N_1]$; and (ii) $\mathsf{wt}(r_{i_2} \oplus u_{i_1,i_2}) \geq 2\kappa/5$ with probability $1 - 2^{-\Omega(\kappa)}$ for all $i_1 \in [N_1]$ and $i_2 \in [N_2]$.

Distributions and Lemma for Tracing. Given $\rho = \sigma_1 \cdots \sigma_t \in \{0,1,\star\}^t$ for $t \le \kappa$, we associated ρ with a distribution and write $r \leftarrow \rho$ for

$$r = s_1 \cdots s_\kappa, \quad \begin{cases} s_i = \sigma_i, & \text{if } i \le t \text{ and } \sigma_i \in \{0,1\}; \\ s_i \leftarrow \{0,1\}, & \text{if } i > t \text{ or } \sigma_i = \star. \end{cases}$$

Our tracing algorithm follows the description in the technical overview, except $u^*_{i_1,i_2}$ is sampled from ρ_{i_1,i_2} instead of being fixed. The distributions ρ_{i_1,i_2} are found iteratively. We rely on the following result:

Lemma 1 (implicit in [29, Section 8.1]). *There is an algorithm* $\mathsf{Learn}^B(u, 1^{1/\delta})$ *that given* $u \in \{0,1\}^\kappa$, $\delta > 0$, *and oracle access to a randomized algorithm* B *with bit output, makes* $\mathrm{poly}(\kappa, 1/\delta)$ *calls to* B *and runs in additional time* $\mathrm{poly}(\kappa, 1/\delta)$. *Its output* $\rho \in \{0,1,\star\}^\kappa$ *satisfies the following conditions:*

- *Each symbol of* ρ *is either the corresponding symbol in* u, *or is* $\star$.
- *The number of* $\star$*'s in* ρ *is no greater than* $2\kappa/5$.

Moreover, for all B *and* $\delta > 0$,

$$\Pr\left[\begin{matrix} u \leftarrow \{0,1\}^\kappa \\ \rho \leftarrow \mathsf{Learn}^B(u, 1^{1/\delta}) \end{matrix} : \Pr_{r \leftarrow \rho}[B(r) = 1] \ge \Pr_{r \leftarrow \{0,1\}^\kappa}[B(r) = 1] - \delta\right] = 1 - 2^{-\Omega(\kappa)}.$$

For completeness, we present a proof in Appendix A. We remark that [15,24] solves the same problem using similar techniques in incomparable parameter regimes that are insufficient for our application.

Tracing Algorithm. Given a decoder D and a distribution $\mathcal{D}$ over traitor tracing ciphertexts and encapsulated keys, we write

$$\varepsilon_D(\mathcal{D}) = \Pr\Big[b \leftarrow \{0,1\},\, k_0 \leftarrow \mathcal{K},\, (\mathsf{ct}, k_1) \leftarrow \mathcal{D} : D(\mathsf{ct}, k_b) = b\Big] - \frac{1}{2}.$$

Recall that $\mathcal{K}$ is the key space, cf. Section 2.2. For brevity, we represent $\mathcal{D}$ by $(j_1; r_1, \ldots, r_{N_2})$ used in TB-PLBE.

The algorithm $\mathsf{Trace}^D(\mathsf{pk}, \mathsf{tk}, 1^{1/\varepsilon})$ works as follows:

1. Let $\xi_1 = \frac{\varepsilon}{10N_1}$. Compute estimations $\hat\varepsilon_{i_1}$ of ε_{i_1} within additive error ξ_1 for $i_1 = 0, \ldots, N_1$, where

$$\varepsilon_{i_1} = \varepsilon_D(i_1; r_1, \ldots, r_{N_2}), \qquad r_1, \ldots, r_{N_2} \leftarrow \{0,1\}^\kappa.$$

   Recall that $\varepsilon_D(\cdots)$ is defined by the probability of an efficient experiment minus $\frac{1}{2}$. It suffices to perform $\lceil(\kappa \log 2 + \log(4N_1 + 4))/(2\xi_1^2)\rceil$ independent trials of that experiment and set $\hat\varepsilon_{i_1}$ to be the empirical frequency minus $\frac{1}{2}$.
2. Pick any $i_1^* \in [N_1]$ such that $\hat\varepsilon_{i_1^*-1} - \hat\varepsilon_{i_1^*} \ge 3\xi_1$. If there is no such i_1^*, the algorithm Trace aborts.
3. For every $t < N_2$ and values of $\rho_{i_1^*,1}, \ldots, \rho_{i_1^*,t}$ (to be found later in Step 4), define $B[\rho_{i_1^*,1}, \ldots, \rho_{i_1^*,t}](r)$, which has the values of ρ's hardwired, as

$$\begin{gathered} b \leftarrow \{0,1\}, \quad b' \leftarrow \{0,1\}, \quad k_0 \leftarrow \mathcal{K}, \quad m \leftarrow \mathbb{Z}_p, \quad k_1 = [m]_\mathrm{T}, \\ u^*_{i_1^*,1} \leftarrow \rho_{i_1^*,1}, \ldots, u^*_{i_1^*,t} \leftarrow \rho_{i_1^*,t}, \\ r_{t+1} = r, \quad r_{t+2} \leftarrow \{0,1\}^\kappa, \ldots, r_{N_2} \leftarrow \{0,1\}^\kappa, \\ R = (u^*_{i_1^*,1}, \ldots, u^*_{i_1^*,t}, r_{t+1}, \ldots, r_{N_2}), \\ \text{output } D\big(\mathsf{TBPLBE.Enc}(\mathsf{tpmpk}, (i_1^* + b', m), R), R, k_b\big) \oplus b \oplus b' \oplus 1. \end{gathered}$$

4. Let $\delta = \frac{\varepsilon}{540N_1N_2}$. For $i_2 = 1, \ldots, N_2$, run

$$\rho_{i_1^*,i_2} \leftarrow \mathsf{Learn}^{B[\rho_{i_1^*,1},\ldots,\rho_{i_1^*,i_2-1}]}(u_{i_1^*,i_2}, 1^{1/\delta}).$$

5. Let $\xi_2 = \frac{\varepsilon}{180 N_1 N_2}$. Estimate $\hat{\varepsilon}_{i_1^*,i_2,0}, \hat{\varepsilon}_{i_1^*,i_2,1}$ of $\varepsilon_{i_1^*,i_2,0}, \varepsilon_{i_1^*,i_2,1}$ within additive error ξ_2 for $i_2 = 0, \ldots, N_2$, where

$$\begin{gathered}\varepsilon_{i_1^*,i_2,b'} = \varepsilon_D(i_1^* - 1 + b'; u_{i_1^*,1}^*, \ldots, u_{i_1^*,i_2}^*, r_{i_2+1}, \ldots, r_{N_2}),\\ u_{i_1^*,1}^* \leftarrow \rho_{i_1^*,1}, \ldots, u_{i_1^*,i_2}^* \leftarrow \rho_{i_1^*,i_2}, \quad r_{i_2+1} \leftarrow \{0,1\}^\kappa, \ldots, r_{N_2} \leftarrow \{0,1\}^\kappa.\end{gathered}$$

   They are computed with $\lceil (\kappa \log 2 + \log(8N_2 + 8))/(2\xi_2^2) \rceil$ independent trials.
6. Pick any $i_2^* \in [N_2]$ such that $(\hat{\varepsilon}_{i_1^*,i_2^*-1,0} - \hat{\varepsilon}_{i_1^*,i_2^*-1,1}) - (\hat{\varepsilon}_{i_1^*,i_2^*,0} - \hat{\varepsilon}_{i_1^*,i_2^*,1}) \geq 5\xi_2$. If there is no such i_2^*, the algorithm Trace aborts.
7. Output (i_1^*, i_2^*) as a traitor.

Tracing Security. We prove the following theorem.

Theorem 1. *Assuming* TBPLBE *being a TB-PLBE secure under adaptively chosen $j_1, r_1, \ldots, r_{N_2}$ and semi-adaptively chosen m (cf. Section 2.3), our traitor tracing scheme is secure (cf. Section 2.2).*

Our proof uses the following lemmas which will be proved later:

Lemma 2. *Assuming* TBPLBE *being a TB-PLBE secure under adaptively chosen $j_1, r_1, \ldots, r_{N_2}$ and semi-adaptively chosen m, in the tracing security game, $\varepsilon_{N_1} \leq \frac{\varepsilon(\lambda)}{2}$ with probability $1 - \lambda^{-\omega(1)}$.*

Lemma 3. *Assuming* TBPLBE *being a TB-PLBE secure under adaptively chosen $j_1, r_1, \ldots, r_{N_2}$ and semi-adaptively chosen m, in the tracing security game, $\varepsilon_{i_1^*,N_2,0} - \varepsilon_{i_1^*,N_2,1} \leq \frac{\varepsilon(\lambda)}{20N_1}$ with probability $1 - \lambda^{-\omega(1)}$, where i_1^* is the index found in Step 2 of* Trace.

It should be noted that in the tracing security game, $\varepsilon_{N_1}, \varepsilon_{i_1^*,N_2,0}, \varepsilon_{i_1^*,N_2,1}$ depend on the random coins of $\mathcal{A}$, and more importantly, those used to set up the traitor tracing scheme in the security game, so they are random variables (not constants) even when $\mathcal{A}, \varepsilon, \lambda$ are fixed. Therefore, $\varepsilon_{N_1} \geq \frac{\varepsilon(\lambda)}{2}$ and $\varepsilon_{i_1^*,N_2,0} - \varepsilon_{i_1^*,N_2,1} \leq \frac{\varepsilon(\lambda)}{20N_1}$ are events (i.e., probabilistic) and the above lemmas bound their probabilities.

Lemma 2 corresponds to the claim in the introduction that ε_{N_1} is negligible, and Lemma 3, $\varepsilon_{i_1^*,N_2,0} - \varepsilon_{i_1^*,N_2,1}$. They are indeed negligible with overwhelming probability, but we only need the weakened version as stated in those lemmas.

Proof (Theorem 1). It suffices to prove the following three claims:

- Claim 1: The probability (of the conjunction event) that $\varepsilon_0 \geq \varepsilon(\lambda)$ and Trace aborts at Step 2 is $\lambda^{-\omega(1)}$.[7]
- Claim 2: The probability that Trace aborts at Step 6 is $\lambda^{-\omega(1)}$.
- Claim 3: Let i_1^* and i_2^* be the indices found in Steps 2 and 6, then $\mathsf{sk}_{i_1^*,i_2^*}$ is queried by the adversary in the tracing security game with probability $1 - 2^{-\Omega(\kappa)}\,\mathrm{poly}(N)$ (i.e., user (i_1^*, i_2^*) is not honest).[8]

Let GoodEst be the event that all estimations are within the prescribed additive errors:

$$\begin{aligned} |\hat{\varepsilon}_{i_1} - \varepsilon_{i_1}| &\leq \xi_1 && \text{for all } i_1 = 0, \ldots, N_1,\\ |\hat{\varepsilon}_{i_1^*,i_2,b'} - \varepsilon_{i_1^*,i_2,b'}| &\leq \xi_2 && \text{for all } i_2 = 0, \ldots, N_2 \text{ and } b' = 0, 1. \end{aligned}$$

By the Chernoff bound, the union bound, and how the numbers of trials are set, we have $\Pr[\mathsf{GoodEst}] \geq 1 - 2^{-\kappa}$. We proceed to prove the claims.

[7] As $\kappa = \omega(\log \lambda)$ and $N = \mathrm{poly}(\lambda)$, any statistical error $2^{-\Omega(\kappa)}$ is absorbed by $\lambda^{-\omega(1)}$ when combined with a computational argument, and thus omitted in such case.

[8] Claim 3 does not care about whether $\varepsilon_0 \geq \varepsilon(\lambda)$.

Proof of Claim 1. By Lemma 2 and our choice of ξ_1, with probability $1-\lambda^{-\omega(1)}$,

$$\max_{i_1\in[N_1]}\{\varepsilon_{i_1-1}-\varepsilon_{i_1}\}\geq\frac{1}{N_1}\sum_{i_1\in[N_1]}(\varepsilon_{i_1-1}-\varepsilon_{i_1})=\frac{\varepsilon_0-\varepsilon_{N_1}}{N_1}\geq\frac{\varepsilon(\lambda)-\frac{\varepsilon(\lambda)}{2}}{N_1}=5\xi_1,$$

when $\varepsilon_0\geq\varepsilon(\lambda)$. Then, GoodEst implies

$$\max_{i_1\in[N_1]}\{\hat{\varepsilon}_{i_1-1}-\hat{\varepsilon}_{i_1}\}\geq 5\xi_1-2\xi_1=3\xi_1.$$

This proves Claim 1.

Proof of Claim 2. GoodEst implies

$$\varepsilon_{i_1^*-1}-\varepsilon_{i_1^*}\geq\hat{\varepsilon}_{i_1^*-1}-\hat{\varepsilon}_{i_1^*}-2\xi_1=\xi_1.$$

Note that $\varepsilon_{i_1^*,0,b'}=\varepsilon_{i_1^*-1+b'}$. Together with Lemma 3 and our choice of ξ_1,ξ_2, with probability $1-\lambda^{-\omega(1)}$,

$$\begin{aligned}&\max_{i_2\in[N_2]}\{(\varepsilon_{i_1^*,i_2-1,0}-\varepsilon_{i_1^*,i_2-1,1})-(\varepsilon_{i_1^*,i_2,0}-\varepsilon_{i_1^*,i_2,1})\}\\&\geq\frac{1}{N_2}\big((\varepsilon_{i_1^*,0,0}-\varepsilon_{i_1^*,0,1})-(\varepsilon_{i_1^*,N_2,0}-\varepsilon_{i_1^*,N_2,1})\big)\geq\frac{\xi_1-\frac{\varepsilon(\lambda)}{20N_1}}{N_2}=9\xi_2.\end{aligned}$$

Again with GoodEst, we have

$$\max_{i_2\in[N_2]}\{(\hat{\varepsilon}_{i_1^*,i_2-1,0}-\hat{\varepsilon}_{i_1^*,i_2-1,1})-(\hat{\varepsilon}_{i_1^*,i_2,0}-\hat{\varepsilon}_{i_1^*,i_2,1})\}\geq 9\xi_2-4\xi_2=5\xi_2.$$

This proves Claim 2.

Proof of Claim 3. Let GoodLearn be the event that for all $i_2\in[N_2]$ such that $\mathsf{sk}_{i_1^*,i_2}$ is not queried by the adversary,

$$\Pr_{r\leftarrow\rho_{i_1^*,i_2}}\big[B[\rho_{i_1^*,1},\ldots,\rho_{i_1^*,i_2-1}](r)=1\big]\geq\Pr_{r\leftarrow\{0,1\}^\kappa}\big[B[\rho_{i_1^*,1},\ldots,\rho_{i_1^*,i_2-1}](r)=1\big]-\delta.$$

Note that if $\mathsf{sk}_{i_1^*,i_2}$ is not queried, then $u_{i_1^*,i_2}$ is independent of the tracing security game until Learn is invoked with it, and therefore, Lemma 1 applies. By a union bound over Lemma 1, we have $\Pr[\mathsf{GoodLearn}]=1-2^{-\Omega(\kappa)}\operatorname{poly}(N)$.

Following the definition of B (with $t=i_2-1$) in Step 3 of Trace and that of $\varepsilon_{i_1^*,i_2-1,b'}$ in Step 5, and applying the law of total probability over b',

$$\begin{aligned}&\Pr_{r\leftarrow\{0,1\}^\kappa}\big[B[\rho_{i_1^*,1},\ldots,\rho_{i_1^*,i_2-1}](r)=1\big]\\&=\frac{1}{2}\Pr\big[D\big(\mathsf{TBPLBE.Enc}(\mathsf{tpmpk},(i_1^*+0,m),R),R,k_b\big)\oplus b\oplus 0\oplus 1=1\big]\\&\quad+\frac{1}{2}\Pr\big[D\big(\mathsf{TBPLBE.Enc}(\mathsf{tpmpk},(i_1^*+1,m),R),R,k_b\big)\oplus b\oplus 1\oplus 1=1\big]\\&=\frac{1}{2}\Pr\big[D\big(\mathsf{TBPLBE.Enc}(\mathsf{tpmpk},(i_1^*,m),R),R,k_b\big)=b\big]\\&\quad+\frac{1}{2}\big(1-\Pr\big[D\big(\mathsf{TBPLBE.Enc}(\mathsf{tpmpk},(i_1^*+1,m),R),R,k_b\big)=b\big]\big)\\&=\frac{1}{2}\varepsilon_{i_1^*,i_2-1,0}+\frac{1}{2}\big(1-\varepsilon_{i_1^*,i_2-1,1}\big).\end{aligned}$$

Similarly, considering $t=i_2$ and $\varepsilon_{i_1^*,i_2,b'}$,

$$\Pr_{r\leftarrow\rho_{i_1^*,i_2}}\big[B[\rho_{i_1^*,1},\ldots,\rho_{i_1^*,i_2-1}](r)=1\big]=\frac{1}{2}\varepsilon_{i_1^*,i_2,0}+\frac{1}{2}\big(1-\varepsilon_{i_1^*,i_2,1}\big).$$

Therefore,

$$\begin{aligned}\Pr_{r\leftarrow\rho_{i_1^*,i_2}}\big[B[\rho_{i_1^*,1},\dots,\rho_{i_1^*,i_2-1}](r)=1\big]-\Pr_{r\leftarrow\{0,1\}^\kappa}\big[B[\rho_{i_1^*,1},\dots,\rho_{i_1^*,i_2-1}](r)=1\big]\\ =\frac{-1}{2}\big((\varepsilon_{i_1^*,i_2-1,0}-\varepsilon_{i_1^*,i_2-1,1})-(\varepsilon_{i_1^*,i_2,0}-\varepsilon_{i_1^*,i_2,1})\big).\end{aligned}$$

GoodLearn thus implies that for all $i_2\in[N_2]$ such that $\mathsf{sk}_{i_1^*,i_2}$ is not queried by the adversary,

$$(\varepsilon_{i_1^*,i_2-1,0}-\varepsilon_{i_1^*,i_2-1,1})-(\varepsilon_{i_1^*,i_2,0}-\varepsilon_{i_1^*,i_2,1})\le 2\delta.$$

Together with GoodEst, for all $i_2\in[N_2]$ such that $\mathsf{sk}_{i_1^*,i_2}$ is not queried by the adversary,

$$(\hat\varepsilon_{i_1^*,i_2-1,0}-\hat\varepsilon_{i_1^*,i_2-1,1})-(\hat\varepsilon_{i_1^*,i_2,0}-\hat\varepsilon_{i_1^*,i_2,1})\le 2\delta+4\xi_2<5\xi_2,$$

i.e., except with probability $2^{-\Omega(\kappa)}\operatorname{poly}(N)$, such i_2 cannot be chosen as i_2^* by Trace. This proves Claim 3 and thus Theorem 1. □

Proving Lemmas. To prove Lemma 2 and Lemma 3, we will use the following trick of advantage sign correction:

Lemma 4 ([7, Exercise 2.22(a)]). *Suppose the tuple of a distinguisher D and two distributions $\mathcal{D}_0,\mathcal{D}_1$ follows some joint distribution. Let*

$$\varepsilon:=\Pr_{x\leftarrow\mathcal{D}_0}[D(x)=1]-\Pr_{x\leftarrow\mathcal{D}_1}[D(x)=1]$$

be the signed *advantage of D against $\mathcal{D}_0,\mathcal{D}_1$, which itself is a random variable (because $D,\mathcal{D}_0,\mathcal{D}_1$ are randomized). Consider*

$$\widetilde{b}\leftarrow\{0,1\},\quad \widetilde{x}\leftarrow\mathcal{D}_{\widetilde{b}},\quad \widetilde{c}\leftarrow D(\widetilde{x}),\quad \widetilde{D}(x):=\widetilde{c}\oplus\widetilde{b}\oplus D(x),$$
$$\widetilde{\varepsilon}:=\Pr_{x\leftarrow\mathcal{D}_0}[\widetilde{D}(x)=1]-\Pr_{x\leftarrow\mathcal{D}_1}[\widetilde{D}(x)=1],$$

then $\mathbb{E}[\widetilde{\varepsilon}]=\mathbb{E}[\varepsilon^2]$.

In our reduction algorithm, ε in Lemma 4 is the signed advantage of the decoder D against certain ciphertext distributions $(\mathcal{D}_0,\mathcal{D}_1)$ used by Trace, and we want to prove that ε is negligible with overwhelming probability. However, if we directly use D as *the* distinguisher, depending on the sampling of $D,\mathcal{D}_0,\mathcal{D}_1$, the realization of ε could be positive or negative, causing cancellation in $\mathbb{E}[\varepsilon]$, the advantage of the reduction algorithm. The $\widetilde{\ }$ components estimate the sign of ε with one trial, and $\widetilde{D}$ is an attempted correction of D, which is *the* distinguisher used by the reduction algorithm and immune from cancellation.

We are now ready to present our proofs.

Proof (Lemma 2). Let $\mathcal{A}$ be an efficient adversary against tracing security. We construct the following efficient $\mathcal{B}$ against TB-PLBE security:

- $\mathcal{B}$ launches $\mathcal{A}$, receives 1^N from it, picks κ,N_1,N_2 as specified by the traitor tracing scheme, samples

  $$m_0\leftarrow\mathbb{Z}_p,\quad m_1\leftarrow\mathbb{Z}_p,\quad u_{i_1,i_2}\leftarrow\{0,1\}^\kappa\quad\text{for } i_1\in[N_1],i_2\in[N_2],$$

  sends $1^{N_1},1^{N_2},m_0,m_1$ to TB-PLBE game, and receives back tpmpk. It sends $\mathsf{pk}=\mathsf{tpmpk}$ to $\mathcal{A}$. Here, m_0,m_1 sent to TB-PLBE game are part of the challenge plaintexts.
- When, and only when, $\mathcal{A}$ queries for sk_{i_1,i_2}, the adversary $\mathcal{B}$ queries TB-PLBE game and sends the key to $\mathcal{A}$.
- When $\mathcal{A}$ outputs a decoder D, the adversary $\mathcal{B}$ samples $r_1,\dots,r_{N_2}\leftarrow\{0,1\}^\kappa$, sends N_2,N_2 as the rest of the challenge plaintexts and $r_1,\dots,r_{N_2}$ as the challenge attribute, and receives back tpct.

- $\mathcal{B}$ samples and computes

$$\begin{aligned}\widetilde{m}_0, \widetilde{m}_1 \leftarrow \mathbb{Z}_p, \quad \widetilde{b} \leftarrow \mathbb{Z}_p, \quad \widetilde{r}_1, \dots, \widetilde{r}_{N_2} \leftarrow \{0,1\}^{\kappa},\\ \widetilde{\mathsf{tpct}} \leftarrow \mathsf{TBPLBE.Enc}\big(\mathsf{tpmpk}, (N_1, \widetilde{m}_{\widetilde{b}}), (\widetilde{r}_1, \dots, \widetilde{r}_{N_2})\big),\\ \widetilde{c} \leftarrow D(\widetilde{\mathsf{tpct}}, \widetilde{r}_1, \dots, \widetilde{r}_{N_2}, [\widetilde{m}_1]_{\mathrm{T}}),\\ c \leftarrow D(\mathsf{tpct}, r_1, \dots, r_{N_2}, [m_1]_{\mathrm{T}}).\end{aligned}$$

It outputs $\widetilde{c} \oplus \widetilde{b} \oplus c$.

By Lemma 4, the advantage of $\mathcal{B}$ is $\mathbb{E}[\varepsilon_{N_1}^2]$, which must be $\lambda^{-\omega(1)}$ by TB-PLBE security. It follows that in the tracing security game with $\mathcal{A}$,

$$\begin{aligned}\Pr\left[\varepsilon_{N_1} \geq \frac{\varepsilon(\lambda)}{2}\right] &\leq \Pr\left[\varepsilon_{N_1}^2 \geq \frac{(\varepsilon(\lambda))^2}{4}\right]\\ &= \frac{4}{(\varepsilon(\lambda))^2} \cdot \frac{(\varepsilon(\lambda))^2}{4} \cdot \Pr\left[\varepsilon_{N_1}^2 \geq \frac{(\varepsilon(\lambda))^2}{4}\right]\\ &\leq \frac{(\varepsilon(\lambda))^2}{4} \cdot \mathbb{E}[\varepsilon_{N_1}^2] = \lambda^{-\omega(1)}.\end{aligned}$$

□

Proof (Lemma 3). The reduction is similar to that in the previous proof, with the following changes:

- The selective part of the challenge plaintexts are m, m.
- After $\mathcal{A}$ outputs D, the reduction computes $1/\varepsilon(\lambda)$, runs Trace to obtain i_1^* and $\rho_{i_1^*,i_2}$'s, samples $u_{i_1^*,i_2}^* \leftarrow \rho_{i_1^*,i_2}$ for $i_2 \in [N_2]$, and sends $i_1^* - 1, i_1^*$ as the challenge plaintexts and $u_{i_1^*,1}^*, \dots, u_{i_1^*,N_2}^*$ as the challenge attribute.
- The $\widetilde{\cdot}$ components sampled by the reduction are

$$\begin{aligned}&\widetilde{m} \leftarrow \mathbb{Z}_p && \text{instead of } \widetilde{m}_0, \widetilde{m}_1,\\ &\widetilde{u}_{i_1^*,i_2}^* \leftarrow \rho_{i_1^*,i_2} && \text{instead of } \widetilde{r}_{i_2},\\ &(i_1^* - 1 + \widetilde{b}, \widetilde{m}) \text{ in } \widetilde{\mathsf{tpct}} && \text{instead of } (N_1, \widetilde{m}_{\widetilde{b}}).\end{aligned}$$

We verify the constraints of TB-PLBE. The constraints of sk_{i_1,i_2} for all $i_1 \neq i_1^*$ and all i_2 are satisfied as

$$f_{i_1}(i_1^* - 1, m) = f_{i_1}(i_1^*, m) = \begin{cases} m, & \text{if } i_1 > i_1^*;\\ 0, & \text{if } i_1 \leq i_1^* - 1.\end{cases}$$

The constraint of $\mathsf{sk}_{i_1^*,i_2}$ for each i_2,

$$0 = P\big((u_{i_1^*,1}^*, \dots, u_{i_1^*,N_2}^*), (i_2, u_{i_1^*,i_2})\big) = \begin{cases} 1, & \text{if } \mathrm{wt}(u_{i_1^*,i_2}^* \oplus u_{i_1^*,i_2}) \geq 2\kappa/5;\\ 0, & \text{otherwise;}\end{cases}$$

holds with probability $1 - 2^{-\Omega(\kappa)}$ by Lemma 1 and a standard Chernoff bound.

The reduction checks the constraints (for both non-$\widetilde{\cdot}$ and $\widetilde{\cdot}$ values) and aborts if any of them is violated. By the analysis above, the probability of aborting is $2^{-\Omega(\kappa)} \operatorname{poly}(N) = \lambda^{-\omega(1)}$, which we denote by ε'. By Lemma 4, the reduction algorithm has advantage $\mathbb{E}[(\varepsilon_{i_1^*,N_2,0} - \varepsilon_{i_1^*,N_2,1})^2] - \varepsilon'$, which is $\lambda^{-\omega(1)}$ by TB-PLBE security. Let C be a polynomial upper bound[9] of $\frac{20N_1}{\varepsilon(\lambda)}$, then

$$\begin{aligned}\Pr\left[\varepsilon_{i_1^*,N_2,0} - \varepsilon_{i_1^*,N_2,1} \geq \frac{\varepsilon(\lambda)}{20N_1}\right] &\leq \Pr\left[(\varepsilon_{i_1^*,N_2,0} - \varepsilon_{i_1^*,N_2,1})^2 \geq \frac{1}{C^2}\right]\\ &= C^2 \cdot \frac{1}{C^2} \cdot \Pr\left[(\varepsilon_{i_1^*,N_2,0} - \varepsilon_{i_1^*,N_2,1})^2 \geq \frac{1}{C^2}\right]\\ &\leq C^2\big(\mathbb{E}[(\varepsilon_{i_1^*,N_2,0} - \varepsilon_{i_1^*,N_2,1})^2] - \varepsilon' + \varepsilon'\big) = \lambda^{-\omega(1)}.\end{aligned}$$

□

[9] N_1 is a random variable due to the random coins of $\mathcal{A}$, so it is impossible to write N_1 outside probability or expectation. For non-uniform security we may assume N_1 is fixed for every λ, yet it is better to present the more general proof.

4 Building Attribute-Based Functional Encryption

This section builds TB-PLBE scheme promised in Section 3.1. At the core is an attribute-based functional encryption (AB-FE) for predicate $P : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ and quadratic function class

$$\mathcal{F}^{\mathrm{quad}}_{\ell_1,\ell_2} = \left\{ f^{\mathrm{quad}}_{\mathbf{f}} : \mathbb{Z}_p^{\ell_1} \times \mathbb{Z}_p^{\ell_2} \to \mathbb{Z}_p, (\mathbf{z}_1, \mathbf{z}_2) \mapsto (\mathbf{z}_1 \otimes \mathbf{z}_2)\mathbf{f}^{\top} \;\middle|\; \mathbf{f} \in \mathbb{Z}_p^{\ell_1 \ell_2} \right\} \tag{1}$$

with $\mathcal{Z} = \mathbb{Z}_p^{\ell_1} \times \mathbb{Z}_p^{\ell_2}$, called AB-QFE. We combine a slightly tweaked version of the FE scheme for quadratic functions (QFE) in [28] and a compatible attribute-based key encapsulation mechanism (AB-KEM) for P. In the following sections, we first introduce the two building blocks in Section 4.1 and Section 4.2; for generality, we will work with general AB-KEM in Section 4.3 where we build our AB-QFE scheme and describe AB-KEM for a certain P (i.e., local (read-once) monotone span program) in Section 4.4 needed for traitor tracing in Section 3. We show how to instantiate our AB-QFE to get TB-PLBE in Section 4.5.

4.1 Building Block: Functional Encryption for Quadratic Functions

A functional encryption scheme for function class $\mathcal{F} = \{f : \mathcal{Z} \to \{0,1\}^*\}$ consists of four p.p.t. algorithms:

- $\mathsf{Setup}(1^\lambda, \mathcal{F}, \mathcal{Z}) \to (\mathsf{mpk}, \mathsf{msk})$: The set-up algorithm takes the security parameter 1^λ and the domains $\mathcal{F}, \mathcal{Z}$ as input, and outputs a master public/secret key pair $(\mathsf{mpk}, \mathsf{msk})$.
- $\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, f) \to \mathsf{sk}$: The key generation algorithm takes mpk, msk, and a function $f \in \mathcal{F}$ as input, and outputs a secret key sk.
- $\mathsf{Enc}(\mathsf{mpk}, z) \to \mathsf{ct}$: The encryption algorithm takes mpk and function input $z \in \mathcal{Z}$ as input, and outputs a ciphertext ct.
- $\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}, f, \mathsf{ct}) \to d$: The decryption algorithm takes $\mathsf{mpk}, \mathsf{sk}, f, \mathsf{ct}$ as input and outputs $d \in \{0,1\}^*$.

Correctness. For all $\lambda \in \mathbb{N}$, $\mathcal{F}$, $\mathcal{Z}$, $f \in \mathcal{F}$, $z \in \mathcal{Z}$, we require

$$\Pr\left[\begin{array}{r} (\mathsf{mpk},\mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{F}, \mathcal{Z}) \\ \mathsf{sk} \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, f) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, z) \end{array} : \mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}, f, \mathsf{ct}) = f(z)\right] = 1.$$

Semi-adaptive Simulation Security. For all p.p.t. stateful $\mathcal{A}$, there exists p.p.t. stateful $(\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{KeyGen}}, \widetilde{\mathsf{Enc}})$ such that

$$\left\{\begin{array}{r} (\mathcal{F},\mathcal{Z}) \leftarrow \mathcal{A}(1^\lambda) \\ (\mathsf{mpk},\mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{F}, \mathcal{Z}) \\ z \leftarrow \mathcal{A}(\mathsf{mpk}) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, z) \\ \textbf{output} \quad \mathcal{A}^{\mathsf{KeyGen}(\mathsf{mpk},\mathsf{msk},\cdot)}(\mathsf{ct}) \end{array}\right\} \approx_c \left\{\begin{array}{r} (\mathcal{F},\mathcal{Z}) \leftarrow \mathcal{A}(1^\lambda) \\ \widetilde{\mathsf{mpk}} \leftarrow \widetilde{\mathsf{Setup}}(1^\lambda, \mathcal{F}, \mathcal{Z}) \\ z \leftarrow \mathcal{A}(\widetilde{\mathsf{mpk}}) \\ \widetilde{\mathsf{ct}} \leftarrow \widetilde{\mathsf{Enc}}() \\ \textbf{output} \quad \mathcal{A}^{\widetilde{\mathsf{KeyGen}}(\cdot,\cdot)}(\widetilde{\mathsf{ct}}) \end{array}\right\},$$

where for each query $f_q \in \mathcal{F}$ made by $\mathcal{A}$, we supply $f_q(z)$ to $\widetilde{\mathsf{KeyGen}}$.

QFE. A functional encryption scheme for quadratic functions (QFE) is an FE computing

$$(\mathbf{z}_1, \mathbf{z}_2, \mathbf{u}) \mapsto (\mathbf{z}_1 \otimes \mathbf{z}_2)\mathbf{f}^{\top} + \mathbf{u}\mathbf{v}^{\top}$$

where $\mathbf{z}_1 \in \mathbb{Z}_p^{\ell_1}, \mathbf{z}_2 \in \mathbb{Z}_p^{\ell_2}, \mathbf{u} \in \mathbb{Z}_p^{\ell_3}$ are the function input, and $\mathbf{f} \in \mathbb{Z}_p^{\ell_1 \ell_2}, \mathbf{v} \in \mathbb{Z}_p^{\ell_3}$ are specified by the function. In this work, we consider QFE implemented using pairing. We let KeyGen, Dec take $(\mathbf{f}, [\mathbf{v}]_2)$ instead of $(\mathbf{f}, \mathbf{v})$ as the description of the function, let Enc take $(\mathbf{z}_1, \mathbf{z}_2, [\mathbf{u}]_1)$ instead of $(\mathbf{z}_1, \mathbf{z}_2, \mathbf{u})$, and only require that Dec output $[(\mathbf{z}_1 \otimes \mathbf{z}_2)\mathbf{f}^{\top} + \mathbf{u}\mathbf{v}^{\top}]_{\mathrm{T}}$. We also let the simulator take the function output encoded in $\mathbb{G}_2$ when simulating a key, which will be convenient for the security proof of AB-FE for quadratic functions.

IPFE. Our construction of QFE is similar to that in [28] and uses an inner-product function encryption (IPFE) scheme which is an FE for $\mathbf{u} \mapsto \mathbf{u}\mathbf{v}^\top$, where $\mathbf{u} \in \mathbb{Z}_p^{\ell_4}$ is the function input and $\mathbf{v} \in \mathbb{Z}_p^{\ell_4}$ is specified by the function. Again, in a group-based scheme, $\mathsf{KeyGen}, \mathsf{Dec}$ takes $[\mathbf{v}]_2$, Enc takes $[\mathbf{u}]_1$, Dec outputs $[\mathbf{u}\mathbf{v}^\top]_\mathrm{T}$, and $\widetilde{\mathsf{KeyGen}}$ takes $[\mathbf{u}\mathbf{v}^\top]_2$. The IPFE in [2] is first proved to be semi-adaptively simulation-secure in [27]. Its parameter sizes are (ignoring constants)

$$|\mathsf{mpk}| = \ell_4|\mathbb{G}_1|, \qquad |\mathsf{ct}| = \ell_4|\mathbb{G}_1|, \qquad |\mathsf{sk}| = |\mathbb{G}_2|.$$

Construction. Suppose k_2-Lin holds in $\mathbb{G}_2$ and bilateral k_{12}-Lin holds. Let IPFE be a semi-adaptively simulation-secure IPFE. Our QFE is as follows:

- $\mathsf{Setup}(1^\lambda, 1^{\ell_1}, 1^{\ell_2}, 1^{\ell_3})$ samples $\mathbf{A}_1 \leftarrow \mathbb{Z}_p^{k_{12}\times\ell_1}$, $\mathbf{A}_2 \leftarrow \mathbb{Z}_p^{k_2\times\ell_2}$, sets the IPFE dimension to $\ell_4 = k_2\ell_1 + k_{12}\ell_2 + \ell_3$, runs $(\mathsf{impk}, \mathsf{imsk}) \leftarrow \mathsf{IPFE.Setup}(1^\lambda, 1^{\ell_4})$, and outputs

  $$\mathsf{mpk} = \big([\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, \mathsf{impk}\big), \qquad \mathsf{msk} = \mathsf{imsk}.$$

- $\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathbf{f}, [\mathbf{v}]_2)$ outputs

  $$\mathsf{sk} = \mathsf{isk} \leftarrow \mathsf{IPFE.KeyGen}\big(\mathsf{imsk}, \big[(\mathbf{A}_1 \otimes \mathbf{I}_{\ell_2})\mathbf{f}^\top,\ (\mathbf{I}_{\ell_1} \otimes \mathbf{A}_2)\mathbf{f}^\top,\ \mathbf{v}\big]_2\big).$$

- $\mathsf{Enc}(\mathsf{mpk}, \mathbf{z}_1, \mathbf{z}_2, [\mathbf{u}]_1)$ samples $\mathbf{s}_1 \leftarrow \mathbb{Z}_p^{k_{12}}$, $\mathbf{s}_2 \leftarrow \mathbb{Z}_p^{k_2}$, run

  $$\mathsf{ict} \leftarrow \mathsf{IPFE.Enc}(\mathsf{impk}, [-\mathbf{s}_1 \otimes \mathbf{z}_2,\ -(\mathbf{s}_1\mathbf{A}_1 + \mathbf{z}_1) \otimes \mathbf{s}_2,\ \mathbf{u}]_1)$$

  and outputs

  $$\mathsf{ct} = \big([\mathbf{s}_1\mathbf{A}_1 + \mathbf{z}_1]_1, [\mathbf{s}_2\mathbf{A}_2 + \mathbf{z}_2]_2, \mathsf{ict}\big),$$

- $\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}, \mathbf{f}, [\mathbf{v}]_2, \mathsf{ct})$ outputs

  $$\big[\underbrace{(\mathbf{s}_1\mathbf{A}_1 + \mathbf{z}_1)}_{\text{in ct}} \otimes \underbrace{(\mathbf{s}_2\mathbf{A}_2 + \mathbf{z}_2)}_{\text{in ct}} \cdot \mathbf{f}^\top\big]_\mathrm{T} \cdot \mathsf{IPFE.Dec}(\mathsf{impk}, \mathsf{isk}, \big[(\mathbf{A}_1 \otimes \mathbf{I}_{\ell_2})\mathbf{f}^\top,\ (\mathbf{I}_{\ell_1} \otimes \mathbf{A}_2)\mathbf{f}^\top,\ \mathbf{v}\big]_2, \mathsf{ict})$$

The correctness is analogous to [28], we defer the details to Appendix B. Its parameter sizes are (ignoring constants)

$$\begin{aligned}
|\mathsf{mpk}| &= \ell_1|\mathbb{G}_1| + (\ell_1 + \ell_2)|\mathbb{G}_2| + |\mathsf{impk}| = (\ell_1 + \ell_2 + \ell_3)|\mathbb{G}_1| + (\ell_1 + \ell_2)|\mathbb{G}_2|,\\
|\mathsf{ct}| &= \ell_1|\mathbb{G}_1| + \ell_2|\mathbb{G}_2| + |\mathsf{ict}| = (\ell_1 + \ell_2 + \ell_3)|\mathbb{G}_1| + \ell_2|\mathbb{G}_2|,\\
|\mathsf{sk}| &= |\mathsf{isk}| = |\mathbb{G}_2|.
\end{aligned}$$

Security. We have the following theorem. The proof is analogous to that for QFE in [28], we defer the details to Appendix B.

Theorem 2. *Assume* IPFE *is semi-adaptively simulation-secure,* k_2*-Lin holds in* $\mathbb{G}_2$*, and bilateral* k_{12}*-Lin holds, our QFE scheme achieves semi-adaptive simulation security.*

4.2 Building Block: Attribute-Based Key Encapsulation Mechanism

We define attribute-based key encapsulation mechanism (AB-KEM) with syntactical properties compatible for constructing AB-FE for quadratic functions. Fix the source groups $\mathbb{G}_1, \mathbb{G}_2$ and the target group $\mathbb{G}_\mathrm{T}$, an AB-KEM for predicate $P : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$ consists of four p.p.t. algorithms:

- $\mathsf{Setup}(1^\lambda, \mathcal{X}, \mathcal{Y}) \to (\mathsf{mpk}, [\mathbf{A}]_2)$: The set-up algorithm takes the security parameter 1^λ and the domains $\mathcal{X}, \mathcal{Y}$ as input, and outputs a master public key mpk and a public matrix $[\mathbf{A}]_2$ with $\mathbf{A} \in \mathbb{Z}_p^{\ell_3\times\ell_5}$.
- $\mathsf{KeyGen}(\mathsf{mpk}, \mathbf{k}, y) \to \mathsf{sk}$: The key generation algorithm takes mpk, a vector $\mathbf{k} \in \mathbb{Z}_p^{\ell_5}$, and $y \in \mathcal{Y}$ as input, and outputs a secret key sk.
- $\mathsf{Enc}(\mathsf{mpk}, \mathbf{s}, x) \to \mathsf{ct}$: The encapsulation algorithm takes a vector $\mathbf{s} \in \mathbb{Z}_p^{\ell_3}$ and $x \in \mathcal{X}$ as input, and outputs a ciphertext ct.
- $\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}, y, \mathsf{ct}, x) \to d$: The decapsulation algorithm takes $\mathsf{mpk}, \mathsf{sk}, y, \mathsf{ct}, x$ as input, and outputs an encapsulated key d.

Correctness. For all $\lambda \in \mathbb{N}$, $\mathcal{X}$, $\mathcal{Y}$, $\mathbf{k} \in \mathbb{Z}_p^{\ell_5}$, $\mathbf{s} \in \mathbb{Z}_p^{\ell_3}$, $x \in \mathcal{X}$, $y \in \mathcal{Y}$ such that $P(x,y) = 1$, we require

$$\Pr\left[\begin{array}{rl} (\mathsf{mpk},[\mathbf{A}]_2) \leftarrow \mathsf{Setup}(1^\lambda,\mathcal{X},\mathcal{Y}) & \\ \mathsf{sk} \leftarrow \mathsf{KeyGen}(\mathsf{mpk},\mathbf{k},y) & : \ \mathsf{Dec}(\mathsf{mpk},\mathsf{sk},y,\mathsf{ct},x) = [\mathbf{sAk}^\top]_\mathrm{T} \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk},\mathbf{s},x) & \end{array}\right] = 1.$$

Adaptive Indistinguishability. For all p.p.t. stateful $\mathcal{A}$, we require

$$\Pr\left[\begin{array}{r} (\mathcal{X},\mathcal{Y}) \leftarrow \mathcal{A}(1^\lambda) \\ (\mathsf{mpk},[\mathbf{A}]_2) \leftarrow \mathsf{Setup}(1^\lambda,\mathcal{X},\mathcal{Y}) \\ \mathbf{s} \leftarrow \mathbb{Z}_p^{\ell_3} \\ x \leftarrow \mathcal{A}^{\mathsf{NewKey}_{\mathsf{kem}}(\cdot)}([\mathbf{sA}]_2) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk},\mathbf{s},x) \\ : \ \ \mathcal{A}^{\mathsf{NewKey}_{\mathsf{kem}}(\cdot)}(\mathsf{ct}) = 1 \end{array}\right] - \Pr\left[\begin{array}{r} (\mathcal{X},\mathcal{Y}) \leftarrow \mathcal{A}(1^\lambda) \\ (\mathsf{mpk},[\mathbf{A}]_2) \leftarrow \mathsf{Setup}(1^\lambda,\mathcal{X},\mathcal{Y}) \\ \mathbf{s} \leftarrow \mathbb{Z}_p^{\ell_3} \\ x \leftarrow \mathcal{A}^{\mathsf{NewKey}_{\$}(\cdot)}([\mathbf{sA}]_2) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk},\mathbf{s},x) \\ : \ \ \mathcal{A}^{\mathsf{NewKey}_{\$}(\cdot)}(\mathsf{ct}) = 1 \end{array}\right]$$

to be negligible under the constraint that $P(x,y_q) = 0$ for all query y_q made by $\mathcal{A}$, where $\mathsf{NewKey}_{\mathsf{kem}}(y_q)$ and $\mathsf{NewKey}_{\$}(y_q)$ run

$$\mathbf{k}_q \leftarrow \mathbb{Z}_p^{\ell_5}, \quad \mathsf{sk}_q \leftarrow \mathsf{KeyGen}(\mathsf{mpk},\mathbf{k}_q,y_q), \quad \begin{cases} \mathsf{kem}_q \leftarrow [\mathbf{sAk}_q^\top]_2, & \text{in } \mathsf{NewKey}_{\mathsf{kem}}; \\ \mathsf{kem}_q \leftarrow \mathbb{G}_2, & \text{in } \mathsf{NewKey}_{\$}; \end{cases}$$

and return $(\mathsf{sk}_q,[\mathbf{Ak}_q^\top]_2,\mathsf{kem}_q)$ to $\mathcal{A}$. The security notion requires that the encapsulated key be pseudorandom even when encoded in $\mathbb{G}_2$, which is stronger than the usual requirement for KEM. This strengthening is for security reduction from AB-FE.

Remark 2. Our formalization basically captures the setting with one key per instance for polynomially many instances and requires that master secret key have special structure. In more detail, mpk here is the public parameter shared among all instances and $\mathbf{k}$ is the master secret key. We will show a concrete construction for local (read-once) monotone span programs in Section 4.4. The construction can be generalized to support a broader class of predicates (see Remark 3).

4.3 AB-FE for Quadratic Functions

We present our AB-FE for quadratic functions (AB-QFE) as defined in (1). In Setup, the functionality $\mathcal{F}^{\mathrm{quad}}_{\ell_1,\ell_2}$ is represented by $(1^{\ell_1},1^{\ell_2})$; in KeyGen and Dec, the function $f^{\mathrm{quad}}_{\mathbf{f}}$ is represented by $\mathbf{f}$ where $\mathbf{f} \in \mathbb{Z}_p^{\ell_1\ell_2}$. In this section, we consider general P, as defined in (1) and Section 4.2, and provide concrete instance for our use.

Construction. Let QFE be the QFE scheme and ABE an AB-KEM for predicate P. Our AB-QFE for P works as follows:

- $\mathsf{Setup}(1^\lambda,1^{\ell_1},1^{\ell_2},\mathcal{X},\mathcal{Y})$ runs

  $$(\mathsf{qmpk},\mathsf{qmsk}) \leftarrow \mathsf{QFE.Setup}(1^\lambda,1^{\ell_1},1^{\ell_2},1^{\ell_3}), \qquad (\mathsf{abmpk},[\mathbf{A}]_2) \leftarrow \mathsf{ABE.Setup}(1^\lambda,\mathcal{X},\mathcal{Y}),$$

  and outputs

  $$\mathsf{mpk} = (\mathsf{abmpk},\mathsf{qmpk}) \quad \text{and} \quad \mathsf{msk} = ([\mathbf{A}]_2,\mathsf{qmsk}).$$

- $\mathsf{KeyGen}(\mathsf{mpk},\mathsf{msk},\mathbf{f},y)$ samples $\mathbf{k} \leftarrow \mathbb{Z}_p^{\ell_5}$, runs

  $$\mathsf{absk} \leftarrow \mathsf{ABE.KeyGen}(\mathsf{abmpk},\mathbf{k},y), \quad \mathsf{qsk} \leftarrow \mathsf{QFE.KeyGen}(\mathsf{qmsk},\mathbf{f},[\mathbf{Ak}^\top]_2),$$

  and outputs

  $$\mathsf{sk} = (\mathsf{absk},\mathsf{qsk},[\mathbf{Ak}^\top]_2).$$

- $\mathsf{Enc}(\mathsf{mpk}, \mathbf{z}_1, \mathbf{z}_2, x)$ samples $\mathbf{s} \leftarrow \mathbb{Z}_p^{\ell_3}$, runs

  $$\mathsf{abct} \leftarrow \mathsf{ABE.Enc}(\mathsf{abmpk}, \mathbf{s}, x), \quad \mathsf{qct} \leftarrow \mathsf{QFE.Enc}(\mathsf{qmpk}, \mathbf{z}_1, \mathbf{z}_2, [\mathbf{s}]_1),$$

  and outputs

  $$\mathsf{ct} = (\mathsf{abct}, \mathsf{qct}).$$

- $\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}, \mathbf{f}, y, \mathsf{ct}, x)$ checks whether $P(x, y) = 0$ and aborts if so. Otherwise, $P(x, y) = 1$, it runs

  $$d_{\mathsf{ABE}} \leftarrow \mathsf{ABE.Dec}(\mathsf{abmpk}, \mathsf{absk}, y, \mathsf{abct}, x), d_{\mathsf{QFE}} \leftarrow \mathsf{QFE.Dec}(\mathsf{qmpk}, \mathsf{qsk}, \mathbf{f}, [\mathbf{A}\mathbf{k}^\top]_2, \mathsf{qct}),$$

  and outputs $d_{\mathsf{QFE}} d_{\mathsf{ABE}}^{-1}$.

Correctness. When $P(x, y) = 1$, the correctness follows from those of ABE and QFE which imply that

$$d_{\mathsf{QFE}} = [(\mathbf{z}_1 \otimes \mathbf{z}_2)\mathbf{f}^\top + \mathbf{s}\mathbf{A}\mathbf{k}^\top]_{\mathrm{T}}, \quad d_{\mathsf{ABE}} = [\mathbf{s}\mathbf{A}\mathbf{k}^\top]_{\mathrm{T}}$$

Efficiency. Our scheme inherits the efficiency from the building blocks:

$$|\mathsf{mpk}| = |\mathsf{abmpk}| + |\mathsf{qmpk}|, \qquad |\mathsf{ct}| = |\mathsf{abct}| + |\mathsf{qct}|, \qquad |\mathsf{sk}| = |\mathsf{absk}| + |\mathsf{qsk}| + \ell_3|\mathbb{G}_2|.$$

Here, ℓ_3 depends on the assumption used by AB-KEM.

Security. We have the following theorem.

Theorem 3. *Assuming* QFE *is semi-adaptively simulation-secure as defined in Section 4.1 and* ABE *achieves security as defined in Section 4.2, our AB-QFE scheme achieves security as defined in Section 2.3.*

Let $(\mathbf{z}_{0,1}, \mathbf{z}_{0,2}, \mathbf{z}_{1,1}, \mathbf{z}_{1,2})$ be the semi-adaptive challenge message, x be the adaptive challenge attribute and $(\mathbf{f}_q, y_q)$ be the q-th query. We prove Theorem 3 via the following game sequence where we write $\eta_{q,b} = (\mathbf{z}_{b,1} \otimes \mathbf{z}_{b,2})\mathbf{f}_q^\top$ for $b \in \{0,1\}$.

- G_0 is the real game, where the keys and the challenge ciphertext are

  $$\begin{aligned} \mathsf{mpk} &= (\mathsf{abmpk}, \mathsf{qmpk}), \\ \mathsf{ct} &= \big(\mathsf{ABE.Enc}(\mathsf{abmpk}, \mathbf{s}, x), \underbrace{\mathsf{QFE.Enc}(\mathsf{qmpk}, \mathbf{z}_{b,1}, \mathbf{z}_{b,2}, [\mathbf{s}]_1)}_{\mathsf{qct}}\big), \\ \mathsf{sk}_q &= \big(\mathsf{ABE.KeyGen}(\mathsf{abmpk}, \mathbf{k}_q, y_q), \underbrace{\mathsf{QFE.KeyGen}(\mathsf{qmpk}, \mathbf{f}_q, [\mathbf{A}\mathbf{k}_q^\top]_2)}_{\mathsf{qsk}_q}, [\mathbf{A}\mathbf{k}_q^\top]_2\big). \end{aligned}$$

- G_1 is identical to G_0, except we use the simulator for QFE to generate QFE components:

  $$\mathsf{mpk} = (\mathsf{abmpk}, \widetilde{\mathsf{qmpk}}), \qquad \mathsf{qct} = \mathsf{QFE}.\widetilde{\mathsf{Enc}}(),$$
  $$\mathsf{qsk}_q = \mathsf{QFE}.\widetilde{\mathsf{KeyGen}}(\mathbf{f}_q, [\mathbf{A}\mathbf{k}_q^\top]_2, [\eta_{q,b} + \mathbf{s}\mathbf{A}\mathbf{k}_q^\top]_2).$$

  We have $\mathsf{G}_0 \approx_c \mathsf{G}_1$ by semi-adaptive simulation security of QFE.

- G_2 is identical to G_1, except we change $\mathbf{s}\mathbf{A}\mathbf{k}_q^\top$ in sk_q to uniformly random $\mu_q \leftarrow \mathbb{Z}_p$ if $\eta_{q,0} \neq \eta_{q,1}$:

  $$\mathsf{qsk}_q = \begin{cases} \mathsf{QFE}.\widetilde{\mathsf{KeyGen}}(\mathbf{f}_q, [\mathbf{A}\mathbf{k}_q^\top]_2, [\eta_{q,b} + \mathbf{s}\mathbf{A}\mathbf{k}_q^\top]_2), & \text{if } \eta_{q,0} = \eta_{q,1}; \\ \mathsf{QFE}.\widetilde{\mathsf{KeyGen}}(\mathbf{f}_q, [\mathbf{A}\mathbf{k}_q^\top]_2, [\eta_{q,b} + \mu_q]_2), & \text{if } \eta_{q,0} \neq \eta_{q,1}. \end{cases}$$

  We have $\mathsf{G}_1 \approx_c \mathsf{G}_2$ by adaptive security of AB-KEM. Roughly speaking, the reduction algorithm receives abmpk, $[\mathbf{A}]_2$, $[\mathbf{s}\mathbf{A}]_2$, and abct from the AB-KEM game. To answer an AB-QFE key query from the adversary, if $\eta_{q,0} = \eta_{q,1}$, the reduction algorithm samples $\mathbf{k}_q \leftarrow \mathbb{Z}_p^{\ell_5}$ and computes sk_q using $\mathsf{abmpk}, [\mathbf{A}]_2, [\mathbf{s}\mathbf{A}]_2, \mathbf{k}_q, \mathbf{f}_q$. Otherwise, it queries the AB-KEM game to obtain $\mathsf{absk}_q, [\mathbf{A}\mathbf{k}_q^\top]_2, \mathsf{kem}_q$, where kem_q is either $[\mathbf{s}\mathbf{A}\mathbf{k}_q^\top]_2$ or random, and computes sk_q with $[\eta_{q,b}]_2 \cdot \mathsf{kem}_q$ as the third argument to $\mathsf{QFE}.\widetilde{\mathsf{KeyGen}}$.

The advantage is 0 in G_2. To see this, note that b only appears in G_2 as

$$\big(\{\eta_{q,b}\}_{\eta_{q,0}=\eta_{q,1}}, \{\eta_{q,b}+\mu_q\}_{\eta_{q,0}\neq\eta_{q,1}}\big) \equiv \big(\{\eta_{q,0}\}_{\eta_{q,0}=\eta_{q,1}}, \{\tilde{\mu}_q\}_{\eta_{q,0}\neq\eta_{q,1}}\big),$$

thus completely hidden.

4.4 AB-KEM for Local (Read-Once) Monotone Span Programs

We describe AB-KEM for local (read-once) monotone span program [10,11]; we redo the scheme and proof in order to fit our syntax and security notion as well as pursue optimal size.

Preliminaries. An m-local (read-once) monotone span program (roMSP) with input length n is specified by $(\mathbf{M},\rho)$, where $\mathbf{M}\in\mathbb{Z}_p^{m\times t}$ and $\rho:[m]\to[n]$ is injective.[10] The predicate for roMSPs of input length n is

$$P_{n,m}^{\mathrm{roMSP}}\big(x,(\mathbf{M},\rho)\big)=\begin{cases}1, & \text{if } \mathbf{e}_1\in\operatorname{span}\{\mathbf{m}_j \mid x_{\rho(j)}=1\};\\ 0, & \text{otherwise;}\end{cases}$$

where $x\in\{0,1\}^n$ and $\mathbf{m}_j$ is the j^{th} row of $\mathbf{M}$. We say x is accepted by $(\mathbf{M},\rho)$ for $P\big(x,(\mathbf{M},\rho)\big)=1$. It is also worth noting that by the tight equivalence between monotone span programs and linear secret sharing schemes (LSSS) [21], m-local roMSPs are equivalent to LSSS where at most m parties have a share and the size of each party's share is at most one.

Construction. The AB-KEM for local roMSPs, denoted by $\mathsf{ABQFEMSP}$, is as follows:

- $\mathsf{Setup}(1^\lambda,1^n)$ samples

  $$\mathbf{A}\leftarrow\mathbb{Z}_p^{k_{12}\times(k_{12}+1)},\quad \mathbf{B}\leftarrow\mathbb{Z}_p^{k_2\times(k_2+1)},\qquad \mathbf{W}_i\leftarrow\mathbb{Z}_p^{(k_{12}+1)\times(k_2+1)}\text{ for } i\in[n],$$

  and outputs

  $$\mathsf{mpk}=\big([\mathbf{A}]_1,\{[\mathbf{AW}_i]_1\}_{i\in[n]},[\mathbf{B}]_2,\{[\mathbf{BW}_i^\top]_2\}_{i\in[n]}\big)\quad\text{and}\quad[\mathbf{A}]_2.$$

- $\mathsf{KeyGen}\big(\mathsf{mpk},\mathbf{k},(\mathbf{M},\rho)\big)$ samples $\mathbf{r}\leftarrow\mathbb{Z}_p^{k_2}$ and $\mathbf{T}'\leftarrow\mathbb{Z}_p^{(t-1)\times(k_{12}+1)}$, sets $\mathbf{T}=\begin{pmatrix}\mathbf{k}\\ \mathbf{T}'\end{pmatrix}$, and outputs

  $$\mathsf{sk}=\Big([\mathbf{r}\cdot\mathbf{B}]_2,\big\{[\mathbf{m}_j\mathbf{T}+\mathbf{r}\cdot\mathbf{BW}_{\rho(j)}^\top]_2\big\}_{j\in[m]}\Big).$$

- $\mathsf{Enc}(\mathsf{mpk},\mathbf{s},x)$ outputs

  $$\mathsf{ct}=\big([\mathbf{s}\cdot\mathbf{A}]_1,\{[\mathbf{s}\cdot\mathbf{AW}_i]_1\}_{x_i=1}\big).$$

- $\mathsf{Dec}(\mathsf{mpk},\mathsf{sk},(\mathbf{M},\rho),\mathsf{ct},x)$ checks whether $P_{n,m}^{\mathrm{roMSP}}\big(x,(\mathbf{M},\rho)\big)=0$ and aborts if so. Otherwise, it finds $\beta_1,\dots,\beta_n\in\mathbb{Z}_p$ such that $\sum_{x_{\rho(j)}=1}\beta_{\rho(j)}\mathbf{m}_j=\mathbf{e}_1$, and outputs

  $$\prod_{x_{\rho(j)}=1}\frac{[\beta_{\rho(j)}\cdot(\overbrace{\mathbf{m}_j\mathbf{T}+\mathbf{rBW}_{\rho(j)}^\top}^{\mathsf{sk}})\cdot(\overbrace{\mathbf{sA}}^{\mathsf{ct}})^\top]_{\mathrm{T}}}{[\beta_{\rho(j)}\cdot\underbrace{\mathbf{rB}}_{\mathsf{sk}}\cdot(\underbrace{\mathbf{sAW}_{\rho(j)}}_{\mathsf{ct}})^\top]_{\mathrm{T}}}.$$

The correctness is straight-forward (cf. [10], more details can be found in Appendix C). The parameter sizes of the scheme are (ignoring constants)

$$|\mathsf{mpk}|=n|\mathbb{G}_1|+n|\mathbb{G}_2|,\qquad \ell_3=\ell_5=1,\qquad |\mathsf{ct}|=\mathsf{wt}(x)|\mathbb{G}_1|,\qquad |\mathsf{sk}|=m|\mathbb{G}_2|,$$

where $n,m,\mathsf{wt}(x)$ are the length of x, the locality of the span program (number of rows of $\mathbf{M}$), and the Hamming weight of x.

[10] It is important that we do *not* assume ρ is the identity map by enlarging $\mathbf{M}$, so that we capture key size dependency in m, the locality. The scheme will be instantiated for κ-local roMSPs ($\kappa\ll n$), which is crucial for the efficiency of our application.

Security. We have the following theorem. The proof basically follows that in [10] and we defer the details to Appendix C.

Theorem 4. *Assume k_2-Lin holds in $\mathbb{G}_2$ and bilateral k_{12}-Lin holds, our AB-KEM for local roMSPs achieves security defined in Section 4.2.*

Remark 3. Our construction is basically a concrete instantiation of [10] and the proof is adapted from it. The adaptation can be generalized to support predicate encoding: only step $\mathsf{G}_4^0 \equiv \mathsf{G}_4^1$ relies on the so-called α-privacy of predicate encoding, other steps, which are irrelevant to the predicate, remain unchanged. Thanks to versatile instantiations of predicate encoding (cf. Appendix A in [10]), this allows us to cover AB-KEM (and thus AB-FE) for arithmetic branching program. Furthermore, many existing dual-system ABE schemes in prime-order pairing groups [20,22,16,23] can be fit into this definition with slight tweaks.

4.5 Threshold Broadcast, Private Linear Broadcast Encryption

Putting Section 4.3 and Section 4.4, we readily have an AB-FE for

$$\mathcal{F}^{\mathrm{quad}}_{\ell_1,\ell_2} \quad \text{and} \quad P^{\mathrm{roMSP}}_{n,m} \tag{2}$$

with

$$|\mathsf{mpk}| = O(\ell_1+\ell_2+n),\, |\mathsf{ct}| = O(\ell_1+\ell_2+n),\, |\mathsf{sk}| = O(m).$$

We can obtain a TB-PLBE scheme (for message space $\mathbb{Z}_p$) as an AB-FE for

$$\mathcal{F}^{\mathrm{comp}}_{N_1} \quad \text{and} \quad P^{\mathrm{tbe}}_{N_2,\kappa} \tag{3}$$

as defined in Section 3.1 by applying the following efficiently computable mappings that reduce (3) to (2) so that

$$\ell_1 = \ell_2 = N_1^{1/2} \quad \text{and} \quad n = 2\kappa N_2,\, m = \kappa.$$

This gives our TB-PLBE with shorter parameters:

$$|\mathsf{mpk}| = O(N_1^{1/2}+\kappa N_2),\, |\mathsf{ct}| = O(N_1^{1/2}+\kappa N_2),\, |\mathsf{sk}| = O(\kappa).$$

Below, we first describe the mappings in general. In **Setting Parameters for Efficiency**, we choose the optimal parameters achieving the desired efficiency.

Function Part. For any $n_1, n_2 \in \mathbb{N}$ satisfying $n_1 n_2 = N_1$, we can perform $\mathcal{F}^{\mathrm{comp}}_{N_1} \mapsto \mathcal{F}^{\mathrm{quad}}_{n_1,n_2}$. For this, we follow [3, Section 6.1] and define

$$\begin{aligned}
\eta_z : &\quad [0,N_1]\times\mathbb{Z}_p \to \mathbb{Z}_p^{n_1}\times\mathbb{Z}_p^{n_2}, &\quad (j_1,m) \mapsto &\begin{cases}(m\mathbf{e}_{j_{11}},\mathbf{e}_{j_{12}}), & \text{if } j_1 < N_1;\\ (\mathbf{0},\mathbf{0}), & \text{if } j_1 = N_1;\end{cases}\\
\eta_f : &\quad [N_1] \to \mathbb{Z}_p^{n_1 n_2}, &\quad i_1 \mapsto &\sum_{0\le j_1 < i_1} \mathbf{e}_{j_{11}}\otimes\mathbf{e}_{j_{12}},
\end{aligned}$$

where $j_{11} \in [n_1]$, $j_{12} \in [n_2]$ satisfy $(j_{11}-1)n_2 + (j_{12}-1) = j_1$ for $j_1 \in [0, N_1-1]$. Note that, for notation convenience, j_1 is the input of η_z and also serves as a general index in the description of η_f. It is straightforward to verify that

$$f^{\mathrm{comp}}_{i_1}(j_1,m) = \mathcal{F}^{\mathrm{quad}}_{\eta_f(i_1)}(\eta_z(j_1,m)).$$

This follows from the fact that

$$\langle \mathbf{e}_{j_{11}}\otimes\mathbf{e}_{j_{12}}, \mathbf{e}_{j'_{11}}\otimes\mathbf{e}_{j'_{12}}\rangle = \begin{cases}1 & \text{if } j_1 = j'_1;\\ 0 & \text{if } j_1 \neq j'_1;\end{cases}$$

where j'_{11}, j'_{12} are defined analogous to j_{11}, j_{12}.

Predicate Part. We will perform $P^{\text{tbe}}_{N_2,\kappa} \mapsto P^{\text{roMSP}}_{2\kappa N_2,\kappa}$. For this, we define

$$\begin{aligned}
\eta_x : \quad & (\{0,1\}^\kappa)^{N_2} \to \{0,1\}^{2\kappa N_2}, \qquad (r_1,\ldots,r_{N_2}) \mapsto (r_1,\bar{r}_1,\ldots,r_{N_2},\bar{r}_{N_2}),\\
\eta_y : \quad & [N_2]\times\{0,1\}^\kappa \to \{\kappa\text{-local roMSP with input length } 2\kappa N_2\},\\
& (i_2,u) \mapsto (\mathbf{M},\rho) \text{ with } \mathbf{M} = \begin{pmatrix} \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_\kappa \end{pmatrix} \in \mathbb{Z}_p^{\kappa\times 2\kappa/5}\\
& \text{where} \quad \mathbf{m}_\theta = (1,\theta,\theta^2,\cdots,\theta^{2\kappa/5-1}),\\
& \qquad \rho(\theta) = 2(i_2-1)\kappa + u_\theta\kappa + \theta, \qquad \forall\theta\in[\kappa].
\end{aligned}$$

The construction of η_y is simply Shamir's secret sharing [25]. We show that

$$P^{\text{tbe}}_{N_2,\kappa}((r_1,\ldots,r_{N_2}),(i_2,u)) = P^{\text{roMSP}}_{2\kappa N_2,\kappa}(\eta_x(r_1,\ldots,r_{N_2}),\eta_y(i_2,u))$$

by proving that $\mathsf{wt}(r_{i_2}\oplus u) \geq 2\kappa/5$ if and only if $\eta_x(r_1,\ldots,r_{N_2})$ is accepted by $\eta_y(i_2,u)$. To see this, first note that

$$\begin{aligned}
(\eta_x(r_1,\ldots,r_{N_2}))_{\rho(\theta)} &= (\overbrace{r_1,\bar{r}_1,\ldots,r_{i_2-1},\bar{r}_{i_2-1}}^{2(i_2-1)\kappa \text{ bits}},r_{i_2},\bar{r}_{i_2},\ldots)_{2(i_2-1)\kappa+u_\theta\kappa+\theta}\\
&= (r_{i_2},\bar{r}_{i_2})_{u_\theta\kappa+\theta} = \begin{cases} (r_{i_2})_\theta, & \text{if } u_\theta = 0;\\ (\bar{r}_{i_2})_\theta, & \text{if } u_\theta = 1;\end{cases}\\
&= (r_{i_2})_\theta \oplus u_\theta.
\end{aligned}$$

By the Vandermonde determinant, any $2\kappa/5$ vectors among $\mathbf{e}_1$ and $\mathbf{m}_\theta$'s are linearly independent, and therefore,

$$\begin{aligned}
\eta_y(i_2,u) \text{ accepts } \eta_x(r_1,\ldots,r_{N_2}) &\iff |\{\theta \mid (\eta_x(r_1,\ldots,r_{N_2}))_{\rho(\theta)} = 1\}| \geq 2\kappa/5\\
&\iff |\{\theta \mid (r_{i_2})_\theta \oplus u_\theta = 1\}| \geq 2\kappa/5\\
&\iff \mathsf{wt}(r_{i_2}\oplus u) \geq 2\kappa/5.
\end{aligned}$$

Transformation. Given those mappings and an AB-FE ABQFEMSP for $\mathcal{F}^{\text{quad}}_{n_1,n_2}$ and $P^{\text{roMSP}}_{2\kappa N_2,\kappa}$, our TBPLBE for $\mathcal{F}^{\text{comp}}_{N_1}$ and $P^{\text{tbe}}_{N_2,\kappa}$ works as follows for $n_1n_2 = N_1$:

- $\mathsf{Setup}(1^\lambda,1^{N_1},1^{N_2},1^\kappa)$ is

$$\mathsf{ABQFEMSP.Setup}(1^\lambda,1^{n_1},1^{n_2},1^{2\kappa N_2});$$

- $\mathsf{KeyGen}(\mathsf{tpmsk},i_1,i_2,u_{i_1,i_2})$ is

$$\mathsf{ABQFEMSP.KeyGen}(\mathsf{tpmsk},\eta_f(i_1),\eta_y(i_2,u_{i_1,i_2}));$$

- $\mathsf{Enc}(\mathsf{tpmpk},j_1,m,r_1,\ldots,r_{N_2})$ is

$$\mathsf{ABQFEMSP.Enc}(\mathsf{tpmpk},\eta_z(j_1,m),\eta_x(r_1,\ldots,r_{N_2}));$$

- $\mathsf{Dec} = \mathsf{ABQFEMSP.Dec}$.

Setting Parameters for Efficiency. By definition, $\mathsf{wt}(\eta_x(r_1,\ldots,r_{N_2})) = \kappa N_2$ and the roMSP $\eta_y(i_2,u)$ is always κ-local. We have TB-PLBE with parameter sizes (ignoring constants)

$$|\mathsf{tpmpk}| = (n_1+n_2+\kappa N_2)|\mathbb{G}_1| + (n_1+n_2+\kappa N_2)|\mathbb{G}_2|, \qquad |\mathsf{tpct}| = (n_1+n_2+\kappa N_2)|\mathbb{G}_1| + n_2|\mathbb{G}_2|, \qquad |\mathsf{tpsk}| = \kappa|\mathbb{G}_2|,$$

where $n_1n_2 = N_1$ and $N_1N_2 = N$. By setting

$$n_1 = n_2 = N^{1/3}\kappa^{1/3} \quad \text{and} \quad N_2 = N^{1/3}\kappa^{-2/3},$$

we obtain

$$|\mathsf{tpmpk}| = N^{1/3}\kappa^{1/3}|\mathbb{G}_1| + N^{1/3}\kappa^{1/3}|\mathbb{G}_2|, \qquad |\mathsf{tpct}| = N^{1/3}\kappa^{1/3}|\mathbb{G}_1| + N^{1/3}\kappa^{1/3}|\mathbb{G}_2|, \qquad |\mathsf{tpsk}| = \kappa|\mathbb{G}_2|.$$

Security. We will need our TB-PLBE to be secure under adaptively chosen $j_1, r_1, \ldots, r_{N_2}$ and semi-adaptively chosen m. The scheme we obtain is already adaptive in $r_1, \ldots, r_{N_2}$ and semi-adaptive in j_1, m. Since $j_1 \in [0, N_1]$ is within a polynomial range, by a standard guessing argument, the scheme is also adaptive in j_1, at a loss of $\frac{1}{N_1+1}$.

References

1. Abdalla, M., Catalano, D., Gay, R., Ursu, B.: Inner-product functional encryption with fine-grained access control. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part III. LNCS, vol. 12493, pp. 467–497. Springer, Heidelberg (Dec 2020). https://doi.org/10.1007/978-3-030-64840-4_16
2. Agrawal, S., Libert, B., Stehlé, D.: Fully secure functional encryption for inner products, from standard assumptions. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part III. LNCS, vol. 9816, pp. 333–362. Springer, Heidelberg (Aug 2016). https://doi.org/10.1007/978-3-662-53015-3_12
3. Baltico, C.E.Z., Catalano, D., Fiore, D., Gay, R.: Practical functional encryption for quadratic functions with applications to predicate encryption. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part I. LNCS, vol. 10401, pp. 67–98. Springer, Heidelberg (Aug 2017). https://doi.org/10.1007/978-3-319-63688-7_3
4. Billet, O., Phan, D.H.: Efficient traitor tracing from collusion secure codes. In: Safavi-Naini, R. (ed.) ICITS 08. LNCS, vol. 5155, pp. 171–182. Springer, Heidelberg (Aug 2008). https://doi.org/10.1007/978-3-540-85093-9_17
5. Boneh, D., Naor, M.: Traitor tracing with constant size ciphertext. In: Ning, P., Syverson, P.F., Jha, S. (eds.) ACM CCS 2008. pp. 501–510. ACM Press (Oct 2008). https://doi.org/10.1145/1455770.1455834
6. Boneh, D., Sahai, A., Waters, B.: Fully collusion resistant traitor tracing with short ciphertexts and private keys. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 573–592. Springer, Heidelberg (May / Jun 2006). https://doi.org/10.1007/11761679_34
7. Boneh, D., Shoup, V.: A Graduate Course in Applied Cryptography (2015), version 0.2, https://toc.cryptobook.us/
8. Boneh, D., Waters, B.: A fully collusion resistant broadcast, trace, and revoke system. In: Juels, A., Wright, R.N., De Capitani di Vimercati, S. (eds.) ACM CCS 2006. pp. 211–220. ACM Press (Oct / Nov 2006). https://doi.org/10.1145/1180405.1180432
9. Boneh, D., Zhandry, M.: Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 480–499. Springer, Heidelberg (Aug 2014). https://doi.org/10.1007/978-3-662-44371-2_27
10. Chen, J., Gay, R., Wee, H.: Improved dual system ABE in prime-order groups via predicate encodings. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 595–624. Springer, Heidelberg (Apr 2015). https://doi.org/10.1007/978-3-662-46803-6_20
11. Chen, J., Gong, J., Kowalczyk, L., Wee, H.: Unbounded ABE via bilinear entropy expansion, revisited. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part I. LNCS, vol. 10820, pp. 503–534. Springer, Heidelberg (Apr / May 2018). https://doi.org/10.1007/978-3-319-78381-9_19
12. Chen, Y., Vaikuntanathan, V., Waters, B., Wee, H., Wichs, D.: Traitor-tracing from LWE made simple and attribute-based. In: Beimel, A., Dziembowski, S. (eds.) TCC 2018, Part II. LNCS, vol. 11240, pp. 341–369. Springer, Heidelberg (Nov 2018). https://doi.org/10.1007/978-3-030-03810-6_13
13. Chor, B., Fiat, A., Naor, M.: Tracing traitors. In: Desmedt, Y. (ed.) CRYPTO'94. LNCS, vol. 839, pp. 257–270. Springer, Heidelberg (Aug 1994). https://doi.org/10.1007/3-540-48658-5_25
14. Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.: An algebraic framework for Diffie-Hellman assumptions. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 129–147. Springer, Heidelberg (Aug 2013). https://doi.org/10.1007/978-3-642-40084-1_8
15. Etesami, O., Mahloujifar, S., Mahmoody, M.: Computational concentration of measure: Optimal bounds, reductions, and more. In: Chawla, S. (ed.) 31st SODA. pp. 345–363. ACM-SIAM (Jan 2020). https://doi.org/10.1137/1.9781611975994.21
16. Gong, J., Wee, H.: Adaptively secure ABE for DFA from k-Lin and more. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part III. LNCS, vol. 12107, pp. 278–308. Springer, Heidelberg (May 2020). https://doi.org/10.1007/978-3-030-45727-3_10
17. Goyal, R., Koppula, V., Russell, A., Waters, B.: Risky traitor tracing and new differential privacy negative results. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part I. LNCS, vol. 10991, pp. 467–497. Springer, Heidelberg (Aug 2018). https://doi.org/10.1007/978-3-319-96884-1_16
18. Goyal, R., Koppula, V., Waters, B.: Collusion resistant traitor tracing from learning with errors. In: Diakonikolas, I., Kempe, D., Henzinger, M. (eds.) 50th ACM STOC. pp. 660–670. ACM Press (Jun 2018). https://doi.org/10.1145/3188745.3188844

19. Goyal, R., Quach, W., Waters, B., Wichs, D.: Broadcast and trace with N^{ε} ciphertext size from standard assumptions. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part III. LNCS, vol. 11694, pp. 826–855. Springer, Heidelberg (Aug 2019). https://doi.org/10.1007/978-3-030-26954-8_27
20. Ishai, Y., Wee, H.: Partial garbling schemes and their applications. In: Esparza, J., Fraigniaud, P., Husfeldt, T., Koutsoupias, E. (eds.) ICALP 2014, Part I. LNCS, vol. 8572, pp. 650–662. Springer, Heidelberg (Jul 2014). https://doi.org/10.1007/978-3-662-43948-7_54
21. Karchmer, M., Wigderson, A.: On span programs. In: Proceedings of Structures in Complexity Theory. pp. 102–111 (1993)
22. Kowalczyk, L., Wee, H.: Compact adaptively secure ABE for NC^1 from k-Lin. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part I. LNCS, vol. 11476, pp. 3–33. Springer, Heidelberg (May 2019). https://doi.org/10.1007/978-3-030-17653-2_1
23. Lin, H., Luo, J.: Succinct and adaptively secure ABE for ABP from k-lin. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part III. LNCS, vol. 12493, pp. 437–466. Springer, Heidelberg (Dec 2020). https://doi.org/10.1007/978-3-030-64840-4_15
24. Mahloujifar, S., Mahmoody, M.: Can adversarially robust learning leverage computational hardness? CoRR **abs/1810.01407** (2018), http://arxiv.org/abs/1810.01407
25. Shamir, A.: How to share a secret. Communications of the Association for Computing Machinery **22**(11), 612–613 (Nov 1979)
26. Waters, B.: Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (Aug 2009). https://doi.org/10.1007/978-3-642-03356-8_36
27. Wee, H.: Attribute-hiding predicate encryption in bilinear groups, revisited. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017, Part I. LNCS, vol. 10677, pp. 206–233. Springer, Heidelberg (Nov 2017). https://doi.org/10.1007/978-3-319-70500-2_8
28. Wee, H.: Functional encryption for quadratic functions from k-lin, revisited. In: Pass, R., Pietrzak, K. (eds.) TCC 2020, Part I. LNCS, vol. 12550, pp. 210–228. Springer, Heidelberg (Nov 2020). https://doi.org/10.1007/978-3-030-64375-1_8
29. Zhandry, M.: New techniques for traitor tracing: Size $N^{1/3}$ and more from pairings. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part I. LNCS, vol. 12170, pp. 652–682. Springer, Heidelberg (Aug 2020). https://doi.org/10.1007/978-3-030-56784-2_22

Appendix

A Proof of Lemma 1

$\mathsf{Learn}^B(u, 1^{1/\delta})$ works as follows:

1. Let ρ_0 be the empty string and $d = \frac{\delta}{2\kappa}$.
2. For $i = 1, \ldots, \kappa$:
   (a) Compute an estimate $\hat{\delta}_i$ of
   $$\delta_i = \Pr_{r \leftarrow \rho_{i-1} u_i}[B(r) = 1] - \Pr_{r \leftarrow \rho_{i-1}\star}[B(r) = 1]$$
   within additive error d. More precisely, for each of the two probabilities, perform $\lceil 2(\kappa \log 2 + \log 4\kappa)/d^2 \rceil$ independent trials and set $\hat{\delta}_i$ to be the empirical frequency. Let Y_i (resp. $\hat{Y}_i$) be the random variable[11] indicating whether $\delta_i \geq 0$ (resp. $\hat{\delta}_i \geq -d$).
   (b) Let $\rho_i \leftarrow \begin{cases} \rho_{i-1} u_i, & \text{if } \hat{Y}_i = 1; \\ \rho_{i-1}\star, & \text{otherwise.} \end{cases}$
3. Output $\rho \leftarrow \begin{cases} \rho_\kappa, & \text{if the number of } \star\text{'s in } \rho_\kappa \text{ is at most } 2\kappa/5; \\ u, & \text{otherwise.} \end{cases}$

Analysis. By definition, each symbol in ρ is either the corresponding symbol in u or $\star$, and the number of $\star$'s in ρ is at most $2\kappa/5$.

To prove

$$\Pr\left[\begin{matrix} u \leftarrow \{0,1\}^\kappa \\ \rho \leftarrow \mathsf{Learn}^B(u, 1^{1/\delta}) \end{matrix} : \Pr_{r \leftarrow \rho}[B(r) = 1] \geq \Pr_{r \leftarrow \{0,1\}^\kappa}[B(r) = 1] - \delta\right] = 1 - 2^{-\Omega(\kappa)},$$

we will show

$$\Pr[\rho = \rho_\kappa] = 1 - 2^{-\Omega(\kappa)}, \text{ and}$$
$$\Pr\left[\Pr_{r \leftarrow \rho_\kappa}[B(r) = 1] \geq \Pr_{r \leftarrow \{0,1\}^\kappa}[B(r) = 1] - \delta\right] = 1 - 2^{-\Omega(\kappa)},$$

where the probability is taken over $u \leftarrow \{0,1\}^\kappa$ and the randomness of Learn. Let GoodEst be the event that $|\delta_i - \hat{\delta}_i| \leq d$ for all $i \in [\kappa]$. By the Chernoff bound, the union bound, and how the number of trials is set, $\Pr[\mathsf{GoodEst}] \geq 1 - 2^{-\kappa}$.

By definition, we have $\hat{Y}_i \hat{\delta}_i \geq -d$. Also, GoodEst implies $\delta_i - \hat{\delta}_i \geq -d$, from which it follows that

$$\begin{aligned} \Pr_{r \leftarrow \rho_\kappa}[B(r) = 1] - \Pr_{r \leftarrow \{0,1\}^\kappa}[B(r) = 1] &= \sum_{i=1}^{\kappa} \left(\Pr_{r \leftarrow \rho_i}[B(r) = 1] - \Pr_{r \leftarrow \rho_{i-1}}[B(r) = 1] \right) \\ &= \sum_{i=1}^{\kappa} \hat{Y}_i \delta_i = \sum_{i=1}^{\kappa} \hat{Y}_i \hat{\delta}_i + \sum_{i=1}^{\kappa} \hat{Y}_i (\delta_i - \hat{\delta}_i) \geq \kappa(-d) + \kappa(-d) = -2d\kappa = -\delta. \end{aligned}$$

Note that $\mathsf{GoodEst} \wedge (Y_i = 1)$ implies $\hat{Y}_i = 1$, and we have

$$\Pr[\rho = \rho_\kappa] \geq \Pr\left[\sum_{i=1}^{\kappa} \hat{Y}_i \geq 2\kappa/5\right] \geq \Pr\left[\mathsf{GoodEst} \wedge \left(\sum_{i=1}^{\kappa} Y_i \geq 2\kappa/5\right)\right] \geq 1 - \Pr[\neg\mathsf{GoodEst}] - \Pr\left[\sum_{i=1}^{\kappa} Y_i < 2\kappa/5\right].$$

Since $\Pr[\neg\mathsf{GoodEst}] \leq 2^{-\kappa}$, it remains to show

$$\Pr\left[\sum_{i=1}^{\kappa} Y_i < 2\kappa/5\right] = 2^{-\Omega(\kappa)}.$$

[11] Y_i is only used for analysis and need not be efficiently computable.

Let $X_i = Y_1 + \cdots + Y_i - \frac{i}{2}$. For all $\rho_{i-1} \in \{0,1,\star\}^{i-1}$, we have

$$\Pr_{u_i \leftarrow \{0,1\}}\left[\Pr_{r\leftarrow \rho_{i-1}u_i}[B(r)=1] - \Pr_{r\leftarrow\rho_{i-1}\star}[B(r)=1] \ge 0\right] \ge \frac{1}{2},$$

from which it follows that

$$\mathbb{E}[X_i \mid X_{i-1},\cdots,X_1,X_0] = \mathbb{E}\left[X_{i-1}+Y_i-\frac{1}{2} \middle| X_{i-1},\cdots,X_1,X_0\right] = X_{i-1} + \mathbb{E}\left[Y_i - \frac{1}{2} \middle| X_{i-1},\cdots,X_1,X_0\right] \ge X_{i-1},$$

i.e., $X_0,X_1,\ldots,X_\kappa$ is a submartingale. Together with $|X_i - X_{i-1}| = \left|Y_i - \frac{1}{2}\right| \le \frac{1}{2}$, we have by Azuma's inequality

$$\Pr\left[\sum_{i=1}^{\kappa} Y_i < 2\kappa/5\right] = \Pr[X_\kappa < -\kappa/10] \le \exp\left(\frac{-(-\kappa/10)^2}{2\cdot(1/2)^2\cdot\kappa}\right) = 2^{-\Omega(\kappa)}.$$

This completes the proof.

B More Details of QFE in Section 4.1

Correctness. Correctness follows from:

$$\begin{aligned}&\mathsf{IPFE.Dec}(\mathsf{impk},\mathsf{isk},\left[(\mathbf{A}_1\otimes\mathbf{I}_{\ell_2})\mathbf{f}^\top,\ (\mathbf{I}_{\ell_1}\otimes\mathbf{A}_2)\mathbf{f}^\top,\ \mathbf{v}\right]_2,\mathsf{ict})\\ =&\left[-(\mathbf{s}_1\otimes\mathbf{z}_2)(\mathbf{A}_1\otimes\mathbf{I}_{\ell_2})\mathbf{f}^\top - ((\mathbf{s}_1\mathbf{A}_1+\mathbf{z}_1)\otimes\mathbf{s}_2)(\mathbf{I}_{\ell_1}\otimes\mathbf{A}_2)\mathbf{f}^\top + \mathbf{u}\mathbf{v}^\top\right]_T\\ =&\left[(\mathbf{z}_1\otimes\mathbf{z}_2)\mathbf{f}^\top + \mathbf{u}\mathbf{v}^\top - (\mathbf{s}_1\mathbf{A}_1+\mathbf{z}_1)\otimes(\mathbf{s}_2\mathbf{A}_2+\mathbf{z}_2)\cdot\mathbf{f}^\top\right]_T\end{aligned}$$

Simulator. Our (stateful) simulator follows that in [28] and additionally accommodates $\mathbf{v},\mathbf{u}$:

- $\widetilde{\mathsf{Setup}}(1^\lambda,1^{\ell_1},1^{\ell_2},1^{\ell_3})$ samples $\mathbf{A}_1 \leftarrow \mathbb{Z}_p^{k_{12}\times(k_{12}+1)}, \mathbf{A}_2\leftarrow \mathbb{Z}_p^{k_2\times(k_2+1)}$, sets the IPFE dimension to $\ell_4 = k_2\ell_1 + k_{12}\ell_2 + \ell_3$, runs $\widetilde{\mathsf{impk}} \leftarrow \mathsf{IPFE}.\widetilde{\mathsf{Setup}}(1^\lambda,1^{\ell_4})$, and outputs
$$\widetilde{\mathsf{mpk}} = \left([\mathbf{A}_1]_1,[\mathbf{A}_1]_2,[\mathbf{A}_2]_2,\widetilde{\mathsf{impk}}\right).$$
- $\widetilde{\mathsf{Enc}}()$ samples $\tilde{\mathbf{z}}_1\leftarrow\mathbb{Z}_p^{\ell_1}, \tilde{\mathbf{z}}_2 \leftarrow \mathbb{Z}_p^{\ell_2}$, runs $\widetilde{\mathsf{ict}}\leftarrow\mathsf{IPFE}.\widetilde{\mathsf{Enc}}()$, and outputs
$$\widetilde{\mathsf{ct}} = \left([\tilde{\mathbf{z}}_1]_1,[\tilde{\mathbf{z}}_2]_2,\widetilde{\mathsf{ict}}\right).$$
- $\widetilde{\mathsf{KeyGen}}(\mathbf{f},[\mathbf{v}]_2,[d]_2)$ outputs $\widetilde{\mathsf{sk}} = \widetilde{\mathsf{isk}} \leftarrow \mathsf{IPFE}.\widetilde{\mathsf{KeyGen}}\left(\left[(\mathbf{A}_1\otimes\mathbf{I}_{\ell_2})\mathbf{f}^\top,\ (\mathbf{I}_{\ell_1}\otimes\mathbf{A}_2)\mathbf{f}^\top,\ \mathbf{v}\right]_2,[d-(\tilde{\mathbf{z}}_1\otimes\tilde{\mathbf{z}}_2)\mathbf{f}_q^\top]_2\right)$.

Security. We have the following theorem.

Theorem 5. *Assume* IPFE *is semi-adaptively simulation-secure, k_2-Lin holds in $\mathbb{G}_2$, and bilateral k_{12}-Lin holds, our QFE scheme achieves semi-adaptive simulation security.*

Let $(\mathbf{z}_1,\mathbf{z}_2)$ be the semi-adaptive challenge message and $\mathbf{f}_q$ be the q-th query. We prove Theorem 5 via the following game sequence:

- G_0 uses the real scheme, and the adversary receives

$$\begin{aligned}\mathsf{mpk} &= \left([\mathbf{A}_1]_1,[\mathbf{A}_1]_2,[\mathbf{A}_2]_2,\mathsf{impk}\right),\\ \mathsf{ct} &= \left([\mathbf{s}_1\mathbf{A}_1+\mathbf{z}_1]_1,[\mathbf{s}_2\mathbf{A}_2+\mathbf{z}_2]_2,\mathsf{ict}\right), \qquad \mathsf{ict}\leftarrow\mathsf{IPFE.Enc}(\mathsf{impk},[\cdots]_1),\\ \mathsf{sk}_q &= \mathsf{isk}_q \leftarrow \mathsf{IPFE.KeyGen}\left(\mathsf{imsk},\left[(\mathbf{A}_1\otimes\mathbf{I}_{\ell_2})\mathbf{f}_q^\top,\ (\mathbf{I}_{\ell_1}\otimes\mathbf{A}_2)\mathbf{f}_q^\top,\ \mathbf{v}_q\right]_2\right).\end{aligned}$$

- G_1 is identical to G_0 except we switch IPFE to simulation:

$$\begin{aligned} \mathsf{mpk} &= \big([\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, \widetilde{\mathsf{impk}}\big), \\ \mathsf{ct} &= \big([\mathbf{s}_1\mathbf{A}_1 + \mathbf{z}_1]_1, [\mathbf{s}_2\mathbf{A}_2 + \mathbf{z}_2]_2, \widetilde{\mathsf{ict}}\big), \qquad \widetilde{\mathsf{ict}} \leftarrow \mathsf{IPFE}.\widetilde{\mathsf{Enc}}(), \\ \mathsf{sk}_q &= \widetilde{\mathsf{isk}}_q \leftarrow \mathsf{IPFE}.\widetilde{\mathsf{KeyGen}}\Big(\big[(\mathbf{A}_1 \otimes \mathbf{I}_{\ell_2})\mathbf{f}_q^\top,\ (\mathbf{I}_{\ell_1} \otimes \mathbf{A}_2)\mathbf{f}_q^\top,\ \mathbf{v}_q\big]_2, \big[(\mathbf{z}_1 \otimes \mathbf{z}_2)\mathbf{f}_q^\top + \mathbf{u}\mathbf{v}_q^\top - \big((\mathbf{s}_1\mathbf{A}_1 + \mathbf{z}_1) \otimes (\mathbf{s}_2\mathbf{A}_2 + \mathbf{z}_2)\big)\mathbf{f}_q^\top\big]_2\Big). \end{aligned}$$

  We have $\mathsf{G}_0 \approx_c \mathsf{G}_1$ by the semi-adaptive simulation security of IPFE.

- G_2 is identical to G_1 except we replace $\mathbf{s}_1\mathbf{A}_1 + \mathbf{z}_1$ by uniformly random $\tilde{\mathbf{z}}_1$:

$$\begin{aligned} \mathsf{mpk} &= \big([\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, \widetilde{\mathsf{impk}}\big), \\ \mathsf{ct} &= \big([\,\tilde{\mathbf{z}}_1\,]_1, [\mathbf{s}_2\mathbf{A}_2 + \mathbf{z}_2]_2, \widetilde{\mathsf{ict}}\big), \qquad \widetilde{\mathsf{ict}} \leftarrow \mathsf{IPFE}.\widetilde{\mathsf{Enc}}(), \\ \mathsf{sk}_q &= \widetilde{\mathsf{isk}}_q \leftarrow \mathsf{IPFE}.\widetilde{\mathsf{KeyGen}}\Big(\big[(\mathbf{A}_1 \otimes \mathbf{I}_{\ell_2})\mathbf{f}_q^\top,\ (\mathbf{I}_{\ell_1} \otimes \mathbf{A}_2)\mathbf{f}_q^\top,\ \mathbf{v}_q\big]_2, \big[(\mathbf{z}_1 \otimes \mathbf{z}_2)\mathbf{f}_q^\top + \mathbf{u}\mathbf{v}_q^\top - \big(\,\tilde{\mathbf{z}}_1 \otimes (\mathbf{s}_2\mathbf{A}_2 + \mathbf{z}_2)\big)\mathbf{f}_q^\top\big]_2\Big). \end{aligned}$$

  We have $\mathsf{G}_2 \approx_c \mathsf{G}_3$ by the bilateral $\mathrm{MDDH}^1_{k_{12},\ell_1}$ assumption in $\mathbb{G}_1, \mathbb{G}_2$, which is implied by the bilateral k_{12}-Lin assumption. The reduction algorithm takes $[\mathbf{A}_1]_1, [\mathbf{c}]_1, [\mathbf{A}_1]_2, [\mathbf{c}]_2$ as input, where $\mathbf{A}_1 \leftarrow \mathbb{Z}_p^{k_{12}\times\ell_1}$ and $\mathbf{c} \in \mathbb{Z}_p^{\ell_1}$ is either $\mathbf{s}_1\mathbf{A}_1$ for $\mathbf{s}_1 \leftarrow \mathbb{Z}_p^{k_{12}}$ or uniformly random. It samples $\mathbf{A}_2, \mathbf{s}_2$, maintains the IPFE simulator, and sets $\tilde{\mathbf{z}}_1 = \mathbf{c} + \mathbf{z}_1$. To compute the first component of the challenge ciphertext, the reduction algorithm uses $[\mathbf{c}]_1$. To compute the input to $\mathsf{IPFE}.\widetilde{\mathsf{KeyGen}}$, it uses $[\mathbf{A}_1]_2, [\mathbf{c}]_2$.

- G_4 is identical to G_3 except we replace $\mathbf{s}_2\mathbf{A}_2 + \mathbf{z}_2$ by uniformly random $\tilde{\mathbf{z}}_2$:

$$\begin{aligned} \mathsf{mpk} &= \big([\mathbf{A}_1]_1, [\mathbf{A}_1]_2, [\mathbf{A}_2]_2, \widetilde{\mathsf{impk}}\big), \\ \mathsf{ct} &= \big([\tilde{\mathbf{z}}_1]_1, [\,\tilde{\mathbf{z}}_2\,]_2, \widetilde{\mathsf{ict}}\big), \qquad \widetilde{\mathsf{ict}} \leftarrow \mathsf{IPFE}.\widetilde{\mathsf{Enc}}(), \\ \mathsf{sk}_q &= \widetilde{\mathsf{isk}}_q \leftarrow \mathsf{IPFE}.\widetilde{\mathsf{KeyGen}}\Big(\big[(\mathbf{A}_1 \otimes \mathbf{I}_{\ell_2})\mathbf{f}_q^\top,\ (\mathbf{I}_{\ell_1} \otimes \mathbf{A}_2)\mathbf{f}_q^\top,\ \mathbf{v}_q\big]_2, \big[\overbrace{(\mathbf{z}_1 \otimes \mathbf{z}_2)\mathbf{f}_q^\top + \mathbf{u}\mathbf{v}_q^\top}^{d_q} - (\tilde{\mathbf{z}}_1 \otimes \,\tilde{\mathbf{z}}_2\,)\mathbf{f}_q^\top\big]_2\Big). \end{aligned}$$

  We have $\mathsf{G}_3 \approx_c \mathsf{G}_4$ by the $\mathrm{MDDH}^1_{k_2,\ell_2}$ assumption in $\mathbb{G}_2$, which is implied by the k_2-Lin assumption. The reduction is similar to that for showing $\mathsf{G}_2 \approx_c \mathsf{G}_3$. Note that G_4 is the same as using the simulator.

C More Details of AB-KEM in Section 4.4

Correctness. The correctness follows from

$$\prod_{x_{\rho(j)}=1} \frac{[\beta_{\rho(j)} \cdot (\mathbf{m}_j\mathbf{T} + \mathbf{r}\mathbf{B}\mathbf{W}_{\rho(j)}^\top) \cdot (\mathbf{s}\mathbf{A})^\top]_{\mathrm{T}}}{[\beta_{\rho(j)} \cdot \mathbf{r}\mathbf{B} \cdot (\mathbf{s}\mathbf{A}\mathbf{W}_{\rho(j)})^\top]_{\mathrm{T}}} = \Bigg[\sum_{x_{\rho(j)}=1} \beta_{\rho(j)}\mathbf{m}_j\mathbf{T}\mathbf{A}^\top\mathbf{s}^\top\Bigg]_{\mathrm{T}} = \Bigg[\mathbf{e}_1\begin{pmatrix}\mathbf{k}\\ \mathbf{T}'\end{pmatrix}\mathbf{A}^\top\mathbf{s}^\top\Bigg]_{\mathrm{T}} = [\mathbf{s}\mathbf{A}\mathbf{k}^\top]_{\mathrm{T}}.$$

Security. We have the following theorem.

Theorem 6. *Assume k_2-Lin holds in $\mathbb{G}_2$ and bilateral k_{12}-Lin holds, our AB-KEM for local roMSPs achieves security defined in Section 4.2.*

By a standard hybrid argument, it suffices to consider security with only one secret key query. Let $\mathbf{M}$ be the key query, we prove Theorem 6 via the following game sequence.

- G_0^γ is the AB-KEM security game with $\mathsf{NewKey}_{\mathsf{kem}}$ (if $\gamma = 0$) or $\mathsf{NewKey}_\$$ (if $\gamma = 1$), where the adversary receives

$$\begin{aligned} \mathsf{mpk} &= \big([\mathbf{A}]_1, \big\{[\mathbf{A}\mathbf{W}_i]_1\big\}_{i\in[n]}, [\mathbf{B}]_2, \big\{[\mathbf{B}\mathbf{W}_i^\top]_2\big\}_{i\in[n]}\big), && [\mathbf{A}]_2, [\mathbf{s}\mathbf{A}]_2, \\ \mathsf{sk} &= \Big([\mathbf{r}\mathbf{B}]_2, \big\{\big[\mathbf{m}_j\mathbf{T} + \mathbf{r}\mathbf{B}\mathbf{W}_{\rho(j)}^\top\big]_2\big\}_{j\in[m]}\Big), && [\mathbf{s}\mathbf{A}\mathbf{k}^\top + \gamma\mu]_2, \\ \mathsf{ct} &= \big([\mathbf{s}\mathbf{A}]_1, \{[\mathbf{s}\mathbf{A}\mathbf{W}_i]_1\}_{x_i=1}\big). \end{aligned}$$

- G_1^γ is identical to G_0^γ except we change all $\mathbf{sA}$ to $\mathbf{c} \leftarrow \mathbb{Z}_p^{k_{12}+1}$:

$$\begin{aligned}
\mathsf{mpk} &= \big([\mathbf{A}]_1, \{[\mathbf{AW}_i]_1\}_{i\in[n]}, [\mathbf{B}]_2, \{[\mathbf{BW}_i^\top]_2\}_{i\in[n]}\big), && [\mathbf{A}]_2, [\mathbf{c}]_2,\\
\mathsf{sk} &= \Big([\mathbf{rB}]_2, \big\{\big[\mathbf{m}_j\mathbf{T} + \mathbf{rBW}_{\rho(j)}^\top\big]_2\big\}_{j\in[m]}\Big), && [\mathbf{c}\,\mathbf{k}^\top + \gamma\mu]_2,\\
\mathsf{ct} &= \big([\mathbf{c}]_1, \{[\mathbf{c}\,\mathbf{W}_i]_1\}_{x_i=1}\big).
\end{aligned}$$

$\mathsf{G}_0^\gamma \approx_c \mathsf{G}_1^\gamma$ follows from the bilateral k_{12}-Lin assumption in $\mathbb{G}_1, \mathbb{G}_2$. Roughly speaking, the reduction algorithm receives $[\mathbf{A}]_1, [\mathbf{c}]_1, [\mathbf{A}]_2, [\mathbf{c}]_2$ as a bilateral k_{12}-Lin challenge ($\mathbf{c}$ is either $\mathbf{sA}$ or random), samples $\mathbf{W}_i, \mathbf{B}, \mathbf{k}, \mathbf{r}, \mathbf{T}'$, and can compute all the components sent to the adversary.

- G_2^γ is identical to G_1^γ except we change $\mathbf{rB}$ to $\mathbf{d} \leftarrow \mathbb{Z}_p^{k_2+1}$:

$$\begin{aligned}
\mathsf{mpk} &= \big([\mathbf{A}]_1, \{[\mathbf{AW}_i]_1\}_{i\in[n]}, [\mathbf{B}]_2, \{[\mathbf{BW}_i^\top]_2\}_{i\in[n]}\big), && [\mathbf{A}]_2, [\mathbf{c}]_2,\\
\mathsf{sk} &= \Big([\mathbf{d}]_2, \big\{\big[\mathbf{m}_j\mathbf{T} + \mathbf{d}\,\mathbf{W}_{\rho(j)}^\top\big]_2\big\}_{j\in[m]}\Big), && [\mathbf{ck}^\top + \gamma\mu]_2,\\
\mathsf{ct} &= \big([\mathbf{c}]_1, \{[\mathbf{cW}_i]_1\}_{x_i=1}\big).
\end{aligned}$$

We have $\mathsf{G}_1^\gamma \approx_c \mathsf{G}_2^\gamma$, analogous to $\mathsf{G}_0^\gamma \approx_c \mathsf{G}_1^\gamma$, except we only need k_2-Lin in $\mathbb{G}_2$ instead of bilateral k_{12}-Lin.

- G_3^γ is identical to G_2^γ except $\mathbf{c}$ (resp. $\mathbf{d}$) is uniformly random outside the row span of $\mathbf{A}$ (resp. $\mathbf{B}$). We have $\mathsf{G}_2^\gamma \approx_s \mathsf{G}_3^\gamma$.
- G_4^γ is identical to G_3^γ except we compute $\mathbf{a}^\perp$ (resp. $\mathbf{b}^\perp$) such that $\mathbf{A}(\mathbf{a}^\perp)^\top = \mathbf{0}$, $\mathbf{c}(\mathbf{a}^\perp)^\top = 1$ (resp. $\mathbf{B}(\mathbf{b}^\perp)^\top = \mathbf{0}$, $\mathbf{d}(\mathbf{b}^\perp)^\top = 1$) and change the variables by

$$\begin{aligned}
\mathbf{W}_i &= \widetilde{\mathbf{W}}_i + w_i(\mathbf{a}^\perp)^\top\mathbf{b}^\perp, && \widetilde{\mathbf{W}}_i \leftarrow \mathbb{Z}_p^{(k_{12}+1)\times(k_2+1)}, w_i \leftarrow \mathbb{Z}_p \quad \text{for } i\in[n],\\
\mathbf{k} &= \widetilde{\mathbf{k}} + \xi\mathbf{a}^\perp, && \widetilde{\mathbf{k}} \leftarrow \mathbb{Z}_p^{k_{12}+1}, \qquad \xi \leftarrow \mathbb{Z}_p,\\
\mathbf{T}' &= \widetilde{\mathbf{T}}' + \mathbf{t}^\top\mathbf{a}^\perp, && \widetilde{\mathbf{T}}' \leftarrow \mathbb{Z}_p^{(t-1)\times(k_{12}+1)}, \quad \mathbf{t} \leftarrow \mathbb{Z}_p^{t-1}.
\end{aligned}$$

Write $\widetilde{\mathbf{T}} = \begin{pmatrix}\widetilde{\mathbf{k}}\\ \widetilde{\mathbf{T}}'\end{pmatrix}$, then the components the adversary receives become

$$\begin{aligned}
\mathsf{mpk} &= \big([\mathbf{A}]_1, \{[\mathbf{A}\widetilde{\mathbf{W}}_i]_1\}_{i\in[n]}, [\mathbf{B}]_2, \{[\mathbf{B}\widetilde{\mathbf{W}}_i^\top]_2\}_{i\in[n]}\big), && [\mathbf{A}]_2, [\mathbf{c}]_2,\\
\mathsf{sk} &= \left([\mathbf{d}]_2, \left\{\left[\mathbf{m}_j\widetilde{\mathbf{T}} + \mathbf{d}\widetilde{\mathbf{W}}_{\rho(j)}^\top + \left(\mathbf{m}_j\begin{pmatrix}\xi\\ \mathbf{t}^\top\end{pmatrix} + w_j\right)\mathbf{a}^\perp\right]_2\right\}_{j\in[m]}\right), && [\mathbf{c}\widetilde{\mathbf{k}}^\top + \xi + \gamma\mu]_2,\\
\mathsf{ct} &= \big([\mathbf{c}]_1, \{[\mathbf{c}\widetilde{\mathbf{W}}_i + w_i\mathbf{b}^\perp]_1\}_{x_i=1}\big).
\end{aligned}$$

We have $\mathsf{G}_3^\gamma \equiv \mathsf{G}_4^\gamma$.

Lastly, for all $(\mathbf{M},\rho)$ and x such that $P\big(x,(\mathbf{M},\rho)\big) = 0$, it holds that (see [10, Section A.5])

$$\begin{aligned}
&\big(\{\mathbf{m}_j\mathbf{v}^\top + w_i\}_{j\in[m],x_{\rho(j)}=1}, \{w_i\}_{x_i=1}, \xi, \{\mathbf{m}_j\mathbf{v}^\top + w_i\}_{j\in[m],x_{\rho(j)}=0}\big)\\
\equiv\ &\big(\{\mathbf{m}_j\mathbf{v}^\top + w_i\}_{j\in[m],x_{\rho(j)}=1}, \{w_i\}_{x_i=1}, \xi, \{\tilde{w}_j\}_{j\in[m],x_{\rho(j)}=0}\big)\\
\equiv\ &\big(\{\mathbf{m}_j\mathbf{v}^\top + w_i\}_{j\in[m],x_{\rho(j)}=1}, \{w_i\}_{x_i=1}, \xi+\mu, \{\tilde{w}_j\}_{j\in[m],x_{\rho(j)}=0}\big)\\
\equiv\ &\big(\{\mathbf{m}_j\mathbf{v}^\top + w_i\}_{j\in[m],x_{\rho(j)}=1}, \{w_i\}_{x_i=1}, \xi+\mu, \{\mathbf{m}_j\mathbf{v}^\top + w_i\}_{j\in[m],x_{\rho(j)}=0}\big),
\end{aligned}$$

where $\xi, \mu \leftarrow \mathbb{Z}_p$, $w_i, \tilde{w}_i \leftarrow \mathbb{Z}_p$, $\mathbf{v} = (\xi, \mathbf{t})$ for $\mathbf{t} \leftarrow \mathbb{Z}_p^{t-1}$, which implies $\mathsf{G}_4^0 \equiv \mathsf{G}_4^1$. This readily implies that $\mathsf{G}_0^0 \approx_c \mathsf{G}_0^1$.