Almost Tight Multi-Instance Multi-Ciphertext Identity-Based Encryption on Lattices

Xavier Boyen and Qinyi Li(✉)

Queensland University of Technology, Brisbane, Australia
qinyi.li@hdr.qut.eud.au

Abstract. Boyen and Li [AsiaCrypt, 2016] proposed the first almost tightly secure lattice identity-based encryption scheme in the standard model. The security of such scheme is proved under learning with errors assumption in the single-instance, single-challenge setting. In this work, we show how to extend the Boyen-Li scheme to obtain an almost tight security reduction in the multi-instance, multi-ciphertext setting, in which the security loss incurred is $\mathsf{poly}(\kappa)$ in the security parameter κ and independent of the number of adversarial queries.

1 Introduction

To prove that the security of a cryptosystem is based on some computational problem, we provide a reductionist proof (in a properly defined security model) that states: If there exists an efficient adversary with runtime t that breaks the cryptosystem with non-negligible probability ϵ, then an efficient algorithm can be constructed to solve the computational problem with non-negligible probability $\epsilon' = \epsilon/L$ in time $t' \approx t$, which contradicts the assumed hardness of such computational problem. The parameter $L \geq 1$ measures the tightness of such a reduction proof. L usually can be affected by several factors, including the reductionist proof itself, the security parameter, the number of deployed instance of a cryptosystem, the number of adversarial queries and so on. We say a reductionist proof is tight if L is a small constant, and almost tight if L is a polynomial of the security parameter and independent of other factors. An (almost) tight reduction usually has smaller and fixed L, which allows us to implement the cryptosystem with shorter parameters in a more accurate way. In contrast, the parameter L in loose reductions is often large and depends on some uncontrollable quantities, e.g., the number of adversarial queries and the number of system instances. These quantities are difficult to determine accurately when the cryptosystem is deployed. Once these quantities are increased by adversaries, L could go beyond

X. Boyen—Research supported in part by ARC Discovery Project grant number DP140103885 and ARC Future Fellowship FT140101145 from the Australian Research Council.

B. Preneel and F. Vercauteren (Eds.): ACNS 2018, LNCS 10892, pp. 535–553, 2018.
https://doi.org/10.1007/978-3-319-93387-0_28

some bound fixed by the implementation, obscuring the cryptosystem's security. Therefore, (almost) tight reduction is a desirable feature for cryptosystems.

In [11] the authors propose an almost tightly secure identity-based encryption (IBE) scheme from lattice. Its security is based on the hardness of learning-with-errors (LWE) problem and the security of an instantiated pseudorandom function (PRF). The reduction is *tight* in the sense that the security loss during the reduction is independent of the number of key generation queries, say Q_{key}, made by the adversary. To make the whole reduction tight, a PRF with tight reduction is required. However, the security reduction given by Boyen and Li [11] is within the "single instance, single challenge" (SISC) setting where the adversary is only given one instance of the IBE scheme and one challenge ciphertext to attack. In a more realistic scenario, many instances of an IBE scheme would be deployed and there would be many ciphertexts targeted by an adversary. To model this "multi-instance, multi-ciphertext" setting, the adversary is allowed to see any polynomial number of scheme instances, say N, adaptively make any polynomial number of identity key generation queries, say Q_{key}, and receive any polynomial number of challenge ciphertexts, say Q_{enc}. Generically, via a hybrid argument, if an IBE scheme Π is ϵ secure (meaning that adversary breaks Π with probability ϵ in a defined model) in the SISC setting, then Π is $\epsilon' = \epsilon \cdot N \cdot Q_{\mathsf{enc}}$ secure in the MIMC setting. This security loss (i.e., $N \cdot Q_{\mathsf{enc}}$) could be significant since N and Q_{enc} are controlled by the adversary and, therefore, could be large. So it is preferable to have IBE schemes whose security does not depend on Q_{key}, Q_{enc} and N in the MIMC setting.

The first construction of IBE schemes from bilinear pairings with tight reductions in the MIMC setting was given by Hofheinz et al. [24]. Several subsequent works, e.g., [4,17,19,20], show various improvements in weakening underlying assumptions, computational efficiency and size of parameters. On the other hand, there is no tightly secure IBE scheme in the MIMC setting from lattices.

In this work, we propose the first lattice-based IBE scheme that has almost tight security reduction in the MIMC setting. We start from the almost tightly secure lattice IBE scheme by Boyen and Li [11] (the only known such scheme, albeit in the SISC setting), and extend it to have a tight security reduction in the MIMC setting under the LWE assumption.

1.1 Our Techniques

We first briefly review the proof idea of Boyen-Li IBE scheme. Let C_{PRF} be a Boolean circuit of a secure one-bit output pseudorandom function PRF. In the security reduction, given any identity, a simulator devises two publicly computable matrices $\mathbf{F}_b = [\mathbf{A}|\mathbf{A}\mathbf{R}_{\mathsf{id}}]$ and $\mathbf{F}_{1-b} = [\mathbf{A}|\mathbf{A}\tilde{\mathbf{R}}_{\mathsf{id}} + (1 - 2C_{\mathsf{PRF}}(\mathbf{k}, \mathsf{id}))\,\mathbf{G}]$ in which $b = \mathsf{PRF}(\mathbf{k}, \mathsf{id}) \in \{0, 1\}$, $\mathbf{G}$ is the gadget matrix, and the low-norm matrices $\mathbf{R}_{\mathsf{id}}, \tilde{\mathbf{R}}_{\mathsf{id}}$ are only known to the simulator. For a key generation query on identity id, the simulator uses the $\mathbf{G}$ trapdoor of the matrix $\mathbf{F}_{1-b}$ to sample a decryption key. For the encryption (challenge) query, using its LWE samples, the simulator constructs a challenge ciphertext $\mathbf{c}_b^\top = \mathbf{s}^\top[\mathbf{A}|\mathbf{A}\mathbf{R}_{\mathsf{id}}] + \mathbf{e}^\top$ where $\mathbf{e}$ is correlated with the secret matrix $\mathbf{R}_{\mathsf{id}}$. Since b is pseudorandom (if PRF is

secure), the adversary would attack $\mathbf{c}_b^\top$ with probability $\approx 1/2$, providing non-trivial information for solving the LWE problem.

While this idea works well in the single instance and single ciphertext setting, it runs into issues in the MIMC setting, particularly when we aim for an (almost) tight reduction. Firstly, for, say, N instances of such IBE scheme, we will have to provide N instances of PRF (specified by the key $\mathbf{k}_i$). In order to make the reduction independent of N, we need to, at some point, switch all instances of PRF to random function in a single step (or with $\mathsf{poly}(\kappa)$ steps that only depends on the security parameter κ). It is not known how to achieve this with existing normal PRFs (a straightforward hybrid argument introduces a factor N in the security loss). Secondly, in the Boyen-Li IBE scheme, the noise $\mathbf{e}$ of the challenge ciphertext is setup by using $\mathbf{R}_{\mathsf{id}}$. By adding a small "smoothing" noise to $\mathbf{e}$, Boyen and Li showed that $\mathbf{R}_{\mathsf{id}}$ remains hidden under polynomial LWE modulus (assuming the PRF circuit is in NC^1). If adversary is able to make multiple challenge queries with the same identity or correlated identities, such an information-theoretic argument would not work any more. Because the adversary can gradually learn the information about $\mathbf{R}_{\mathsf{id}}$ from multiple challenge ciphertexts on identities that are the same as/correlated to id, and fail the reduction.

We deal with the two issues as follows. Firstly, recall one-bit output PRFs are sufficient for Boyen-Li IBE scheme. We notice that the single-instance security of a PRF with certain key-homomorphism could be tightly extend to the security in multi-instance setting, as long as different PRF instances do not evaluate the same input. A PRF $\mathsf{PRF} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is key homomorphic if $(\mathcal{K}, \boxplus)$ and $(\mathcal{Y}, +)$ are groups, and given $\mathsf{PRF}(\mathbf{k}, \mathbf{x}), \mathsf{PRF}(\mathbf{k}', \mathbf{x})$, then $\mathsf{PRF}(\mathbf{k} \boxplus \mathbf{k}', \mathbf{x}) = \mathsf{PRF}(\mathbf{k}, \mathbf{x}) + \mathsf{PRF}(\mathbf{k}', \mathbf{x})$. Given an oracle access to $\mathsf{PRF}(\mathbf{k}^*, \cdot)$ one can simulate a PRF with a uniformly random key $\mathbf{k}_i$ by freshly choosing a key $\tilde{\mathbf{k}}_i$ and setting its output as $\mathsf{PRF}(\mathbf{k}^*, \cdot) + \mathsf{PRF}(\tilde{\mathbf{k}}_i, \cdot)$. On the other hand, given an oracle access to a random function $F(\cdot)$, one can simulate a random function as $F'(\cdot) = F(\cdot) + \mathsf{PRF}(\tilde{\mathbf{k}}_i, \cdot)$ if all queries are different. However, only approximate key-homomorphic PRFs from lattices are known which satisfy $\mathsf{PRF}(\mathbf{k} \boxplus \mathbf{k}') = \mathsf{PRF}(\mathbf{k}, \mathbf{x}) + \mathsf{PRF}(\mathbf{k}', \mathbf{x}) + \varepsilon$ for a small error term ε. We can set parameters such that ε barely affects the most significant bits of outputs: with overwhelming probability, $\mathsf{MSB}(\mathsf{PRF}(\mathbf{k} \boxplus \mathbf{k}')) = \mathsf{MSB}(\mathsf{PRF}(\mathbf{k}, \mathbf{x}) + \mathsf{PRF}(\mathbf{k}', \mathbf{x}))$. This idea was used in a very different context, i.e., building distributed PRFs from approximate key-homomorphic PRFs [10].

For the issue of constructing multiple challenge ciphertexts (or answering multiple encryption queries), we use the lossy mode of LWE: embedding an instance of LWE problem into the matrix $\mathbf{A}$ make $\mathbf{s}^\top[\mathbf{A}|\mathbf{A}\mathbf{R}_{\mathsf{id}}] + \mathbf{e}^\top$ statistically lose the information of $\mathbf{s}$. While $\mathbf{s}$ and $\mathbf{e}$ now are independent of the LWE problem that we embedded, we can pick fresh $\mathbf{s}, \mathbf{e}$ for each challenge ciphertext and, thus, eliminate the problem that we have in Boyen-Li IBE scheme. Moreover, while one instance of LWE problem is embedded (trough multiple samples) to all scheme instances (i.e., different matrix $\mathbf{A}$), we can switch half of the challenge ciphertexts (the ones indexed by the bit $b_{\mathsf{id}}^{(j)} = \mathsf{PRF}(\mathbf{k}^{(j)}, \mathsf{id})$ for the scheme

instance j.) to random in a single step. Such an idea stems from the notion of lossy trapdoor function [6,27] and has recently been used in [12,25].

2 Preliminaries

We use PPT to denote "probabilistic polynomial-time". We denote by $x||y$ the concatenation of bit x and y. For a positive integer n, we denote by $[n]$ the set of positive integers no greater than n. We use bold lowercase letters (e.g. $\mathbf{a}$) to denote vectors and bold capital letters (e.g. $\mathbf{A}$) to denote matrices. For a positive integer $q \geq 2$, let $\mathbb{Z}_q$ be the ring of integers modulo q. We denote the group of $n \times m$ matrices in $\mathbb{Z}_q$ by $\mathbb{Z}_q^{n\times m}$. Vectors are treated as column vectors. The transpose of a vector $\mathbf{a}$ (resp. a matrix $\mathbf{A}$) is denoted by $\mathbf{a}^\top$ (resp. $\mathbf{A}^\top$). For $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{B} \in \mathbb{Z}_q^{n\times m'}$, let $[\mathbf{A}|\mathbf{B}] \in \mathbb{Z}_q^{n\times(m+m')}$ be the concatenation of $\mathbf{A}$ and $\mathbf{B}$. We write $\|\mathbf{x}\|_\infty$ for the infinity norm of a vector $\mathbf{x}$. The Euclidean norm of a matrix $\mathbf{R} = \{\mathbf{r}_1, \ldots, \mathbf{r}_m\}$ is denoted by $\|\mathbf{R}\| = \max_i \|\mathbf{r}_i\|$. We denote $\|\mathbf{R}\|_{\mathsf{GS}}$ by the Euclidean norm of the Gram-Schmidt orthogonalization of the column vector of $\mathbf{R}$. The spectral norm of $\mathbf{R}$ is denoted by $s_1(\mathbf{R}) = \sup_{\mathbf{x}\in\mathbb{R}^{m+1}} \|\mathbf{R}\cdot\mathbf{x}\|$. For a security parameter κ, a function $\mathsf{negl}(\kappa)$ is negligible in κ if it is smaller than all polynomial fractions for a sufficiently large κ.

2.1 Randomness Extractor

Let X and Y be two random variables over some finite set S. The statistical distance between X and Y, denoted as $\Delta(X,Y)$, is defined as $\Delta(X,Y) = \frac{1}{2}\sum_{s\in S} |\Pr[X=s] - \Pr[Y=s]|$. Let X_λ and Y_λ be ensembles of random variables indexed by the security parameter λ. X and Y are statistically close if $\Delta(X_\lambda, Y_\lambda) = \mathsf{negl}(\lambda)$. The min-entropy of a random variable X over a set S is defined as $H_\infty(X) = -\log(\max_{s\in S} \Pr[X=s])$. A random variable X has ε-smooth min-entropy at least k, denoted by $H_\infty^\varepsilon(X) \geq k$, if there exists some variable X' such that $\Delta(X,X') \leq \varepsilon$ and $H_\infty(X') \geq k$. We write $H_\infty^{\mathsf{smooth}}(\cdot)$ for some (unspecified) negligible ε.

Definition 1 (Universal Hash Functions). *$\mathcal{H} = \{H : \mathcal{X} \to \mathcal{Y}\}$ is called a family of universal hash functions if for all $x, x' \in \mathcal{X}$, with $x \neq x'$, we have $\Pr[H(x) = H(x')] \leq \frac{1}{|\mathcal{Y}|}$ over the random choice of $H \leftarrow \mathcal{H}$.*

Lemma 1 ([27], Lemma 2.2). *Let X, Y be random variables such that $X \in \{0,1\}^n$ and $\tilde{H}_\infty(X|Y) \geq k$. Let $\mathcal{H} : \{0,1\}^n \to \{0,1\}^\ell$ be a family of universal hash functions where $k \geq \ell + 2\lambda$. It holds that for $H \xleftarrow{\$} \mathcal{H}$ and $r \xleftarrow{\$} \{0,1\}^\ell$, $\Delta((H, H(X), Y), (H, r, Y)) \leq 2^{-\lambda}$.*

Lemma 2 ([1], Lemma 4). *Suppose that $m > (n+1)\log q + \omega(\log n)$ and that $q > 2$ is prime. Let $\mathbf{R}$ be an $m \times k$ matrix chosen uniformly in $\{1,-1\}^{m\times k} \mod q$ where $k = k(n)$ is polynomial in n. Let $\mathbf{A}$ and $\mathbf{B}$ be matrices chosen uniformly in $\mathbb{Z}_q^{n\times m}$ and $\mathbb{Z}_q^{n\times k}$ respectively. Then, for all vectors $\mathbf{w} \in \mathbb{Z}_q^m$, the distribution $(\mathbf{A}, \mathbf{AR}, \mathbf{R}^\top\mathbf{w})$ is statistically close to the distribution $(\mathbf{A}, \mathbf{B}, \mathbf{R}^\top\mathbf{w})$.*

2.2 Lattice Background

Definition 2. *Let a basis* $\mathbf{B} = [\mathbf{b}_1 \mid \ldots \mid \mathbf{b}_m] \in (\mathbb{R}^m)^m$ *of linearly independent vectors. The lattice generated by* $\mathbf{B}$ *is defined as* $\Lambda = \{\mathbf{y} \in \mathbb{R}^m : \exists s_i \in \mathbb{Z}, \mathbf{y} = \sum_{i=1}^m s_i \mathbf{b}_i\}$. *For* q *prime,* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, *we define the* m*-dimensional (full-rank) random integer lattice* $\Lambda_q^{\perp}(\mathbf{A}) = \{\mathbf{e} \in \mathbb{Z}^m : \mathbf{Ae} = \mathbf{0} \pmod q\}$.

We denote the discrete Gaussian distribution over a lattice Λ with Gaussian parameter $s > 0$, center $\mathbf{0}$ by $D_{\Lambda,s}$. We refer to [18] for the definition of discrete Gaussian distribution. We recall the following facts of "gadget matrix" [26].

Lemma 3 ([26], Theorem 1). *Let* q *be a prime, and* n, m *be integers with* $m = n \log q$. *There is a fixed full-rank matrix* $\mathbf{G} \in \mathbb{Z}_q^{n \times m}$ *such that the lattice* $\Lambda_q^{\perp}(\mathbf{G})$ *has a publicly known trapdoor matrix* $\mathbf{T_G} \in \mathbb{Z}^{n \times m}$ *with* $\|\mathbf{T_G}\|_{GS} \leq \sqrt{5}$.

Lemma 4 ([9], Lemma 2.1). *There is a deterministic algorithm, denoted* $\mathbf{G}^{-1}(\cdot) : \mathbb{Z}_q^{n \times m} \to \mathbb{Z}^{m \times m}$, *that takes any matrix* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *as input, and outputs the preimage* $\mathbf{G}^{-1}(\mathbf{A})$ *of* $\mathbf{A}$ *such that* $\mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{A}) = \mathbf{A} \pmod q$ *and* $\|\mathbf{G}^{-1}(\mathbf{A})\| \leq \sqrt{m}$.

Lattice Trapdoors. It is shown in [2] how to sample a "nearly" uniform random matrix $\mathbf{A} \in \mathbb{Z}^{n \times m}$ along with a trapdoor matrix $\mathbf{T_A} \in \mathbb{Z}^{m \times m}$ which is a short or low-norm basis of the induced lattice $\Lambda_q^{\perp}(\mathbf{A})$.

Lemma 5. *There is a PPT algorithm* TrapGen *that takes as input integers* $n \geq 1$, $q \geq 2$ *and a sufficiently large* $m = O(n \log q)$, *outputs a matrix* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *and a trapdoor matrix* $\mathbf{T_A} \in \mathbb{Z}^{m \times m}$, *such that* $\mathbf{A} \cdot \mathbf{T_A} = 0 \pmod q$, *the distribution of* $\mathbf{A}$ *is statistically close to the uniform distribution over* $\mathbb{Z}_q^{n \times m}$ *and* $\|\mathbf{T_A}\|_{GS} = O(\sqrt{n \log q})$.

Lemma 6. *Let* n, q, m *be integers with* $m = O(n \log q)$. *Let* $\mathbf{s} \in \mathbb{Z}_q^n$, $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{e} \in \mathbb{Z}^m$. *Given* $\mathbf{y}^\top = \mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top \bmod q$ *and a basis* $\mathbf{T}$ *of* $\Lambda_q^{\perp}(\mathbf{A})$ *such that* $\left\|\mathbf{e}^\top \mathbf{T}\right\|_\infty \leq q/4$, *there is an algorithm* Invert$(\mathbf{y}, \mathbf{A}, \mathbf{T})$ *that outputs* $\mathbf{s}$ *with overwhelming probability.*

We use the following lattice basis sampling algorithms due to [1,16,26].

Lemma 7. *There is an efficient algorithm* SampleLeft *which takes as input a full-rank matrix* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, *a matrix* $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$, *a short basis* $\mathbf{T_A} \in \mathbb{Z}^{m \times m}$, *a Gaussian parameter* s *where* $s > \|\mathbf{T_A}\|_{GS} \cdot \omega(\sqrt{\log 2m})$, *and for* $\mathbf{F} = [\mathbf{A}|\mathbf{B}]$, *outputs a full-rank basis* $\mathbf{T_F}$ *of* $\Lambda_q^{\perp}(\mathbf{F})$ *where the distribution of* $\mathbf{T_F}$ *is statistically close to* $D_{\Lambda_q^{\perp}(\mathbf{F}),s}$ *and* $\|\mathbf{T_F}\|_\infty \leq s\sqrt{2m}$.

Lemma 8. *There is an efficient algorithm* SampleRight *which takes as input* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, *low-norm matrix* $\mathbf{R} \in \mathbb{Z}^{m \times m}$, *non-zero scalar* $h \in \mathbb{Z}_q$, *gadget matrix* $\mathbf{G} \in \mathbb{Z}_q^{n \times m}$, *a Gaussian parameter* s *where* $s > \sqrt{5} \cdot s_1(\mathbf{R}) \cdot \omega(\sqrt{\log m})$, *and for* $\mathbf{F} = [\mathbf{A}|\mathbf{AR} + h\mathbf{G}]$, *outputs a full-rank basis* $\mathbf{T_F}$ *of* $\Lambda_q^{\perp}(\mathbf{F})$ *where the distribution of* $\mathbf{T_F}$ *is statistically close to* $D_{\Lambda_q^{\perp}(\mathbf{F}),s}$ *and* $\|\mathbf{T_F}\|_\infty \leq s\sqrt{2m}$.

Homomorphic Evaluation Algorithm. We adopt the following lemma.

Lemma 9 ([11]). *Let $C : \{0,1\}^\ell \to \{0,1\}$ be a NAND Boolean circuit. Let $\{\mathbf{A}_i = \mathbf{A}\mathbf{R}_i + x_i\mathbf{G} \in \mathbb{Z}_q^{n\times m}\}_{i\in[\ell]}$ be ℓ different matrices correspond to each input wire of C where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, $\mathbf{R}_i \xleftarrow{\$} \{1,-1\}^{m\times m}$, $x_i \in \{0,1\}$ and $\mathbf{G} \in \mathbb{Z}_q^{n\times m}$ is the gadget matrix. There is an efficient deterministic algorithm $\mathsf{Eval}_{\mathsf{BV}}$ that takes as input C and $\{\mathbf{A}_i\}_{i\in[\ell]}$ and outputs a matrix $\mathbf{A}_C = \mathbf{A}\mathbf{R}_C + C(x_1,\ldots,x_\ell)\mathbf{G} = \mathsf{Eval}_{\mathsf{BV}}(C,\mathbf{A}_1,\ldots,\mathbf{A}_\ell)$ where $\mathbf{R}_C \in \mathbb{Z}^{m\times m}$ and $C(x_1,\ldots,x_\ell)$ is the output of C on the arguments $x_1,\ldots,x_\ell$, $s_1(\mathbf{R}_C) \le O(4^d \cdot m^{3/2})$. $\mathsf{Eval}_{\mathsf{BV}}$ runs in time $poly(4^d,\ell,n,\log q)$. Particularly, if C has depth $d = c\log\ell$ for some constant c, i.e. C is in NC^1, we have $s_1(\mathbf{R}_C) \le O(\ell^{2c}\cdot m^{3/2})$.*

Computational Assumptions. We recall the following variant of decision learning with errors assumption.

Definition 3 (Decision LWE). *Let n and q be positive integers. Let χ be a distribution over $\mathbb{Z}_q$. Let $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$ be a secret vector. Define oracles :*

- *$\mathcal{O}_{\mathbf{s}}$: samples $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n$, column vector $\mathbf{e} \leftarrow \chi$; returns $(\mathbf{a}, \mathbf{s}^\top\mathbf{a} + e \bmod q)$.*
- *$\mathcal{O}_{\$}$: samples $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n$, $b \xleftarrow{\$} \mathbb{Z}_q$; returns $(\mathbf{a}, b)$.*

The decision LWE problem, denote $\mathsf{LWE}_{n,q,\chi}$, asks to distinguish between $\mathcal{O}_{\mathbf{S}}$ and $\mathcal{O}_{\$}$. The (decision) LWE assumption says that for an efficient algorithm $\mathcal{A}$, there is a negligible functuon $\mathsf{negl}(\kappa)$ such that

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{LWE}_{n,q,\chi}}(\kappa) = \left|\Pr[\mathcal{A}^{\mathcal{O}_{\mathbf{s}}}(1^\kappa) = 1] - \Pr[\mathcal{A}^{\mathcal{O}_{\$}}(1^\kappa) = 1]\right| \le \mathsf{negl}(\kappa)$$

Notice that the decision LWE problem does not restrict the number of oracle calls (or the number of samples available to $\mathcal{A}$). In the security proof of our IBE scheme, we use this fact to obtain enough samples from a single instance of LWE problem to simulate multiple challenge ciphertexts. Usually, the noise distribution χ is a discrete Gaussian distribution $D_{\mathbb{Z},\alpha q}$ where $\alpha \in (0,1)$ and $\alpha q > 3\sqrt{n}$. For fix dimension n, the modulus-to-noise ratio q/α measures the hardness of LWE problem. The larger the ratio, the easier the LWE problem.

In our construction, we use a variant of LWE problem where the secret is a random matrix $\mathbf{S} \in \mathbb{Z}_q^{n\times h}$ (we choose the noise as vectors where coordinates are independently sampled according to χ). Via a hybrid argument, such a variant is polynomially equivalent to the LWE problem we define above up to a factor of h in the reduction.

2.3 Lossy Mode for LWE

A series of works [3,6,25] show that LWE/LWR problem (with a-priori polynomially bounded number of samples) has a lossy mode in which the samples only reveal partial information of its secret. More precisely, given m LWE samples $\mathbf{y}^\top = \mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top \pmod q$ where $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, if $\mathbf{A}$ is generated in the lossy mode, then $\mathbf{s}$ still has some entropy given $\mathbf{y}$, $\mathbf{A}$. The following lemma states this fact.

Lemma 10 ([3], Lemma B.4). *Let κ be a security parameter. Let n, n', m^*, q, β^*, γ, σ and λ be integers and χ be the LWE error distribution over $\mathbb{Z}_q$ where $\Pr_{x\leftarrow\chi}[|x| \geq \beta^*] \leq \mathsf{negl}(\kappa)$ and $\sigma \geq \beta^*\gamma nm^*$. For random variables $\mathbf{s} \in [-\gamma, \gamma]^n$, $\mathbf{e} \xleftarrow{\$} [\sigma, \sigma]^{m^*}$ and $\mathbf{A} = \mathbf{CB} + \mathbf{F} \pmod q$ where $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n\times n'}$, $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{n'\times m^*}$ and $\mathbf{F} \leftarrow \chi^{n\times m^*}$, we have*

$$H_\infty^{\mathsf{smooth}}(\mathbf{s}|\mathbf{A}, \mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top) \geq H_\infty(\mathbf{s}) - (n' + 2\kappa)\log q$$

The following theorem, which is a direct consequence of Lemma 10, is essential for the security proof of our IBE scheme.

Theorem 1. *Let κ be a security parameter. Let $n, n', m, q, \gamma, \sigma, \lambda$ be integers, q prime, β real, such that $n \geq \kappa$, $m \geq O(n\log q)$. Let χ be the LWE error distribution over $\mathbb{Z}_q$ where $\Pr_{x\leftarrow\chi}[|x| \geq \beta] \leq \mathsf{negl}(\kappa)$. Let $\mathbf{R} \in \mathbb{Z}^{m\times m}$ be a low-norm matrix with $\|\mathbf{R}\|_\infty \leq B$. Assume $n \geq (n' + 2\kappa + \frac{\lambda}{\log q})\frac{\log q}{\log 2\gamma} + \frac{2\kappa}{\log q}$ and $\sigma \geq 2B\beta\gamma nm$. For random variables $\mathbf{s} \in [-\gamma, \gamma]^n$, $\mathbf{e} \xleftarrow{\$} [\sigma, \sigma]^{2m}$ and $\mathbf{A} = \mathbf{CB}+\mathbf{F} \pmod q$ where $\mathbf{C} \xleftarrow{\$} \mathbb{Z}_q^{n\times n'}$, $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{n'\times m}$ and $\mathbf{F} \leftarrow \chi^{n\times m}$ such that given $\mathbf{FR}$, $\mathbf{BR}$ is statistically close to the uniform distribution over $\mathbb{Z}_q^{n'\times m}$, we have*

$$\begin{aligned} H_\infty^{\mathsf{smooth}}(\mathbf{s}|\mathbf{A}, \mathbf{s}^\top[\mathbf{A}|\mathbf{AR}] + \mathbf{e}^\top) &\geq H_\infty(\mathbf{s}) - (n' + 2\kappa)\log q \\ &\geq 2\kappa + \lambda \end{aligned}$$

Proof. The proof follows from the proof of Theorem 7.3, [3]. We can write $[\mathbf{A}|\mathbf{AR}] = \mathbf{CB}^* + \mathbf{F}^*$ where $\mathbf{B}^* = [\mathbf{B}|\mathbf{BR}]$ and $\mathbf{F}^* = [\mathbf{F}|\mathbf{FR}]$. First of all, the statistical distance between the distribution of $\mathbf{B}^*$ and the uniform distribution over $\mathbb{Z}_q^{n'\times 2m}$ is $\mathsf{negl}(\kappa)$. Secondly, we can bound each entry of $\mathbf{F}^*$ by $mB\beta$. Therefore, invoking Lemma 10 with $m^* = 2m$, $\beta^* = B\beta$, $n \geq (n' + 2\kappa + \frac{\lambda}{\log q})\frac{\log q}{\log 2\gamma} + \frac{2\kappa}{\log q}$, $\sigma \geq 2B\beta\gamma nm^2$ and concealing $\mathsf{negl}(\kappa)$ by the term smooth, we have

$$\begin{aligned} H_\infty^{\mathsf{smooth}}(\mathbf{s}|\mathbf{A}, \mathbf{s}^\top[\mathbf{A}|\mathbf{AR}] + \mathbf{e}^\top) &\geq H_\infty(\mathbf{s}) - (n' + 2\kappa)\log q \\ &\geq n\log(2\gamma) - (n' + 2\kappa)\log q \\ &\geq 2\kappa + \lambda \end{aligned}$$

2.4 Identity-Based Encryption

An identity-based encryption (IBE) scheme with identity space $\mathcal{ID}$ and message space $\mathcal{M}$ consists of the following five PPT algorithms:

- $\mathsf{Para}(1^\kappa) \rightarrow \mathsf{pub}$. The public parameter generation algorithm Para takes as input a security parameter κ, and outputs a set of global parameters pub.
- $\mathsf{Setup}(\mathsf{pub}) \rightarrow (\mathsf{mpk}, \mathsf{msk})$. The setup algorithm Setup takes as input pub, and outputs a master public key mpk and a master secret key msk.
- $\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id}) \rightarrow \mathsf{ct}_{\mathsf{id}}$. The key generation algorithm KeyGen takes as input the master public key mpk, the master private key msk, and an identity id, and outputs a user private key $\mathsf{sk}_{\mathsf{id}}$.

- $\mathsf{Encrypt}(\mathsf{mpk}, \mathsf{id}, \mathsf{m}) \to \mathsf{ct}_{\mathsf{id}}$. The encryption algorithm $\mathsf{Encrypt}$ takes as input the master public key mpk, an identity id, and a message m, outputs a ciphertext $\mathsf{ct}_{\mathsf{id}}$.
- $\mathsf{Decrypt}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}}, \mathsf{ct}_{\mathsf{id}}) \to \mathsf{m}$ or $\perp$. The decryption algorithm $\mathsf{Decrypt}$ takes as input the master public key mpk, a private key $\mathsf{sk}_{\mathsf{id}}$ and a ciphertext $\mathsf{ct}_{\mathsf{id}}$, outputs message m or $\perp$.

For correctness, we require that for all κ, all $\mathsf{pub} \leftarrow \mathsf{Para}(1^\kappa)$, all $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(\mathsf{pub})$, all $\mathsf{id} \in \mathcal{ID}$, all $\mathsf{ct}_{\mathsf{id}} \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$, all $\mathsf{m} \in \mathcal{M}$ and for all $\mathsf{ct}_{\mathsf{id}} \leftarrow \mathsf{Encrypt}(\mathsf{mpk}, \mathsf{id}, \mathsf{m})$, $\mathsf{Decrypt}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}}, \mathsf{ct}_{\mathsf{id}})$ outputs m except negligible probability.

Security Definition. The multi-instance, multi-ciphertext security for an IBE scheme $\Pi = (\mathsf{Para}, \mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ is defined through the following security game between a challenger $\mathcal{B}$ and an adversary $\mathcal{A}$.

Initial. $\mathcal{B}$ runs $\mathsf{pub} \leftarrow \mathsf{Para}(1^\kappa)$ and randomly picks $\mathsf{coin} \leftarrow \{0, 1\}$, and gives pub to $\mathcal{A}$. $\mathcal{A}$ selects $N = \mathsf{poly}(\kappa)$. Then $\mathcal{B}$ runs $(\mathsf{mpk}^{(j)}, \mathsf{msk}^{(j)}) \leftarrow \mathsf{Setup}(\mathsf{pub})$ for $j \in [N]$, and gives $\{\mathsf{mpk}^{(j)}\}_{j\in[N]}$ to $\mathcal{A}$.

Query. $\mathcal{A}$ adaptively issues the following two types of queries:

- Key Generation Query. The adversary $\mathcal{A}$ submits $(j \in [N], \mathsf{id} \in \mathcal{ID})$ to the challenger $\mathcal{B}$. $\mathcal{B}$ runs $\mathsf{sk}_{\mathsf{id}}^{(j)} \leftarrow \mathsf{KeyGen}(\mathsf{mpk}^{(j)}, \mathsf{msk}^{(j)}, \mathsf{id})$ and gives $\mathsf{sk}_{\mathsf{id}}^{(j)}$ to $\mathcal{A}$.
- Encryption Query. The adversary submits the k-th encryption query $(k \in [Q_{\mathsf{enc}}], j \in [N], \mathsf{id} \in \mathcal{ID}, \mathsf{m}_0, \mathsf{m}_1 \in \mathcal{M})$ to $\mathcal{B}$. $\mathcal{B}$ runs $\mathsf{ct}_{\mathsf{id},k}^{(j)} \leftarrow \mathsf{Encrypt}(\mathsf{mpk}^{(j)}, \mathsf{id}, \mathsf{m}_{\mathsf{coin}})$ and returns $\mathsf{ct}_{\mathsf{id},k}^{(j)}$ to $\mathcal{A}$. In addition, $\mathcal{A}$ is allowed to submit two encryption queries with same instance index j (but the index k will be different)[1].

Guess. $\mathcal{A}$ outputs $\mathsf{coin}' \in \{0, 1\}$ and it wins if $\mathsf{coin}' = \mathsf{coin}$.

The advantage of $\mathcal{A}$ in wining the game is defined as $\mathsf{Adv}^{\mathsf{IND\text{-}ID\text{-}CPA}}_{\mathcal{A},\Pi,(N,Q_{\mathsf{key}},Q_{\mathsf{enc}})}(\kappa) = |\Pr[\mathsf{coin}' = \mathsf{coin}] - 1/2|$, where Q_{key} and Q_{enc} are the number of key generation queries and encryption queries, respectively. We say that an IBE scheme Π is secure if for all PPT adversary $\mathcal{A}$, there is a negligible function $\mathsf{negl}(\kappa)$ such that $\mathsf{Adv}^{\mathsf{IND\text{-}ID\text{-}CPA}}_{\mathcal{A},\Pi,(N,Q_{\mathsf{key}},Q_{\mathsf{enc}})}(\kappa) \leq \mathsf{negl}(\kappa)$.

2.5 Almost Key-Homomorphic Pseudorandom Functions

Definition 4 (Pseudorandom Functions). *Let κ be the security parameter. A pseudorandom function $\mathsf{PRF} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is an efficiently computable, deterministic function. Let Ω be the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$. We define the advantage of an adversary $\mathcal{A}$ in attacking the PRF as*

$$Adv_{PRF,\mathcal{A}}(\kappa) = \left|\Pr[\mathcal{A}^{PRF(K,\cdot)}(1^\kappa) = 1] - \Pr[\mathcal{A}^{F(\cdot)}(1^\kappa) = 1]\right|$$

[1] This refers to the strong/full adaptive MIMC security [17,24].

where the probability is taken over a uniform choice of key $K \xleftarrow{\$} \mathcal{K}$ *and* $F \xleftarrow{\$} \Omega$*, and the randomness of* $\mathcal{A}$*. We say that* PRF *is secure if for all PPT adversaries* $\mathcal{A}$*,* $\mathsf{Adv}_{\mathsf{PRF},\mathcal{A}}(\kappa) \leq \mathsf{negl}(\kappa)$ *for some negligible function* $\mathsf{negl}(\kappa)$*.*

Definition 5. *A PRF* $\mathsf{PRF} : \mathcal{K} \times \mathcal{X} \to \mathbb{Z}_q$ *is* ε*-almost key-homomorphic if* $(\mathcal{K}, \boxplus)$ *is a group, and for* $\mathbf{k}_1, \mathbf{k}_2 \in \mathcal{K}$*,* $\mathbf{x} \in \mathcal{X}$*, we have*

$$\mathsf{PRF}(\mathbf{k}_1 \boxplus \mathbf{k}_2, \mathbf{x}) = \mathsf{PRF}(\mathbf{k}_1, \mathbf{x}) + \mathsf{PRF}(\mathbf{k}_2, \mathbf{x}) + e$$

where $e \in [0, \varepsilon]$*.*

Let $\mathsf{Prefix}: \mathbb{Z}_p \to \{0,1\}^\ell$ where $\ell \leq \log p$ be a deterministic function that takes as input an element in $\mathbb{Z}_q$ and outputs its binary prefix of length ℓ.

Definition 6. *We say a* ε*-almost key-homomorphic PRF has* p*refix correction with respect to the function* Prefix *if*

$$\mathsf{Prefix}\left(\mathsf{PRF}(\mathbf{k}_1 \boxplus \mathbf{k}_2, \mathbf{x})\right) = \mathsf{Prefix}\left(\mathsf{PRF}(\mathbf{k}_1, \mathbf{x}) + \mathsf{PRF}(\mathbf{k}_2, \mathbf{x})\right)$$

holds with overwhelming probability. Particularly, we say ε*-almost key-homomorphic PRF* PRF *has most-significant-bit correction:*

$$\mathsf{MSB}\left(\mathsf{PRF}(\mathbf{k}_1 \boxplus \mathbf{k}_2, \mathbf{x})\right) = \mathsf{MSB}\left(\mathsf{PRF}(\mathbf{k}_1, \mathbf{x}) + \mathsf{PRF}(\mathbf{k}_2, \mathbf{x})\right)$$

with all but negligible probability where $\mathsf{MSB} : \mathbb{Z}_p \to \{0,1\}$ *be a deterministic function that takes as input an* $\mathbb{Z}_p$*-element and outputs its most significant bit.*

To base our IBE scheme on lattice assumptions with a (almost) tight reduction, we can instantiate the PRF in our construction with the lattice-based almost key-homomorphic PRF by Boneh et al. [10] (BLMR-PRF). Here we recall the construction of BLMR-PRF. Let n, m, p, q be integers where $m = n\lfloor \log q \rfloor$ and $p | q$. Let $\mathbb{Z}_q$-invertible matrices $\mathbf{B}_0, \mathbf{B}_1 \in \{0,1\}^{m \times m}$ be public parameter. For an input $\mathbf{x} = \mathbf{x}[1]\mathbf{x}[2]...\mathbf{x}[\ell] \in \{0,1\}^\ell$, a secret key $\mathbf{k} \leftarrow \mathbb{Z}_q^m$, the BLMR-PRF $\mathsf{PRF}_{\mathsf{BLMR}} : \mathbb{Z}_q^m \times \{0,1\}^\ell \to \mathbb{Z}_p^m$ is defined as

$$\mathsf{PRF}_{\mathsf{BLMR}}(\mathbf{k}, \mathbf{x}) = \left\lfloor \prod_{i=1}^{\ell} \mathbf{B}_{\mathbf{x}[i]} \cdot \mathbf{k} \right\rfloor_p \tag{1}$$

where for any $x \in \mathbb{Z}_q$, the function $\lfloor x \rfloor_p = \lfloor (p/q) \cdot x \rfloor \bmod p$, and it naturally extends to vectors by applying the function to each coordinate of the vector individually. While the output space (of the original description) of BLMR-PRF is $\mathbb{Z}_p^m$, we can always output the first $\mathbb{Z}_p$ coordinate as an input to the function Prefix (and MSB). Assume $2|p$, for $x \in \mathbb{Z}_p$, we define

$$\mathsf{MSB}(x) = \lfloor x \rfloor_2 = \lfloor (2/p) \cdot x \rfloor \bmod 2$$

The 1-almost key-homomorphism of BLMR-PRF was proved in [10] (Theorem 5.5). To make the BMLR-PRF have the most-significant-bit correction property, we can set the parameter p slightly super-polynomial, e.g., $p = n^{\omega(1)}$

(and set up q accordingly), to make sure the noise always properly being rounded off. This fact has already been mentioned in [10] in applying almost key-homomorphic PRFs to obtain distributed PRFs.

Very recently, Libert et al. ([25], Theorem 7) showed that BLMR-PRF has a (almost) tight reduction from non-uniform LWE (NLWE) problem (in the sense that the security loss during the security reduction is independent of the number of PRF queries being made) which in turn has a tight security reduction to LWE problem with certain parameters ([10], Theorem 4.3). These results together demonstrate that for input length ℓ, BLMR-PRF is (almost) tightly secure under the LWE assumption where the modulus-to-noise ratio is $n^{\Omega(\ell)}$.

Similar to the Boyen-Li IBE scheme, using shallow depth almost key-homomorphic PRFs (e.g., the ones can be implemented by NC^1 circuits) will allow us to use polynomial modulus for the IBE scheme (not the PRF itself). BLMR-PRF satisfies this requirement. As it is mentioned in [25], the computation of BLMR-PRF can be divided into two phases, a matrices product followed by rounding an inner-product. The matrices product $\prod_{i=1}^{\ell} \mathbf{B}_{\mathbf{x}[i]}$ can be computed publicly without knowing the secret key. So the actual circuit needed to be evaluated is the "inner-product-then-rounding" circuit which is in NC^1.

3 The Scheme

In our scheme, we require that the same identity is never used for requesting private identity keys from different scheme instances. Such a requirement is natural and essential for the security proof. It is done by appending a unique instance identifier to users' actual identities. A user with identity id' uses the actual identity $\mathsf{id} = \mathsf{ID}||\mathsf{id}'$ for the scheme instance whose identifier is ID.

$\mathsf{Para}(1^\kappa)$. The public parameter generation algorithm does the following.

1. Choose a LWE hardness parameter n', integer $n \geq n'$, integer $m = 2n \log q + \omega(\log n)$, LWE modulo q and integers γ, σ. Set message space $\mathcal{M} = \{0,1\}^\lambda$ for some integer λ.
2. Select an almost key-homomorphic PRF $\mathsf{PRF} : \{0,1\}^t \times \{0,1\}^\ell \to \{0,1\}^r$, where $r = \omega(\log \kappa)$, which has the most-significant-bit correction (as per Definition 6). Set a depth d, NAND Boolean circuit $C_{\mathsf{PRF}}\{0,1\}^t \times \{0,1\}^\ell \to \{0,1\}$ which outputs the most significant bits of the output stings of PRF. That is C_{PRF} computes $\mathsf{MSB}(\mathsf{PRF}(\cdot,\cdot))$.
3. Let $B = O(4^d \cdot m^{3/2})$ (as the bound given in Lemma 9), we choose $s \geq \sqrt{5} \cdot B \cdot \omega(\sqrt{\log 2m})$.
4. Randomly sample a universal hash function $\mathsf{H} : [-\gamma, \gamma]^n \to \{0,1\}^\lambda$ from a family of universal hash functions $\mathcal{H}$.
5. Output the global public parameters

$$\mathsf{pub} = (n, m, q, \gamma, \delta, \lambda, \mathsf{PRF}, C_{\mathsf{PRF}}, \mathsf{H}, s)$$

$\mathsf{Setup}(\mathsf{pub})$. On input pub, the setup algorithm does the following.

1. Select a random key $\mathbf{k} \leftarrow \{0,1\}^t$ for PRF.
2. Run $\mathsf{TrapGen}(n, m, q)$ to generate a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ along with a trapdoor $\mathbf{T_A} \in \mathbb{Z}^{m \times m}$.
3. Choose random matrices $\mathbf{A}_0, \mathbf{A}_1, \mathbf{C}_1, ..., \mathbf{C}_t \leftarrow \mathbb{Z}_q^{n \times m}$.
4. Choose a unique system identifier ID, and output the master public key

$$\mathsf{mpk} = \left(\mathsf{ID}, \mathbf{A}, \mathbf{A}_0, \mathbf{A}_1, \{\mathbf{C}_i\}_{i \in [t]}\right)$$

and master secret key $\mathsf{msk} = (\mathbf{T_A}, \mathbf{k})$.

$\mathsf{Encrypt}(\mathsf{mpk}, \mathsf{id}, \mathsf{m})$. Let $\mathsf{ID}||\mathsf{id} = \mathsf{id}[1]...\mathsf{id}[\ell] \in \{0,1\}^\ell$, the algorithm encrypts $\mathsf{m} \in \{0,1\}^\lambda$ as follows.

1. Compute $\mathbf{A}_{C_{\mathsf{PRF}},\mathsf{id}} = \mathsf{Eval}(C_{\mathsf{PRF}}, \{\mathbf{C}_i\}_{i \in [t]}, \mathsf{id}[1]\mathbf{G}, ..., \mathsf{id}[\ell]\mathbf{G}) \in \mathbb{Z}_q^{n \times m}$.
2. Set $\mathbf{F}_{\mathsf{id},\mu} = [\mathbf{A}|\mathbf{A}_\mu - \mathbf{A}_{C_{\mathsf{PRF}},\mathsf{id}}]$ for $\mu = 0, 1$.
3. Select $\mathbf{x}_0, \mathbf{x}_1 \leftarrow [-\gamma, \gamma]^n, \mathbf{e}_0, \mathbf{e}_1 \leftarrow [-\sigma, \sigma]^{2m}$. Output the ciphertext $\mathsf{ct_{id}} = (\mathbf{c}_0, \mathbf{c}'_0, \mathbf{c}_1, \mathbf{c}'_1)$ where

$$\begin{cases} \mathbf{c}_0 = \mathsf{m} \oplus \mathsf{H}(\mathbf{x}_0) \\ \mathbf{c}_0'^{\top} = \mathbf{x}_0^\top \cdot \mathbf{F}_{\mathsf{id},0} + \mathbf{e}_0^\top \bmod q \end{cases} ; \quad \begin{cases} \mathbf{c}_1 = \mathsf{m} \oplus \mathsf{H}(\mathbf{x}_1) \\ \mathbf{c}_1'^{\top} = \mathbf{x}_1^\top \cdot \mathbf{F}_{\mathsf{id},1} + \mathbf{e}_1^\top \bmod q \end{cases}$$

$\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$. On input $\mathsf{mpk}, \mathsf{msk}$ and an identity id, the algorithm does the following to generate a private key.

1. Compute $\mu = \mathsf{MSB}(\mathsf{PRF}(\mathbf{k}, \mathsf{ID}||\mathsf{id})) \in \{0,1\}$.
2. Compute $\mathbf{A}_{C_{\mathsf{PRF}},\mathsf{id}} = \mathsf{Eval}(C_{\mathsf{PRF}}, \{\mathbf{C}_i\}_{i \in [t]}, \mathsf{id}[1]\mathbf{G}, ..., \mathsf{id}[\ell]\mathbf{G}) \in \mathbb{Z}_q^{n \times m}$.
3. Set $\mathbf{F}_{\mathsf{id},1-\mu} = [\mathbf{A}|\mathbf{A}_{1-\mu} - \mathbf{A}_{C_{\mathsf{PRF}},\mathsf{id}}] \in \mathbb{Z}_q^{n \times 2m}$.
4. Run $\mathsf{SampleLeft}([\mathbf{A}|\mathbf{A}_{1-\mu} - \mathbf{A}_{C_{\mathsf{PRF}},\mathsf{id}}], \mathbf{T_A}, s)$ to get trapdoor $\mathbf{T_{id}} \in \mathbb{Z}^{2m \times 2m}$ for $\mathbf{F}_{\mathsf{id},1-\mu}$.
5. Return $\mathsf{sk_{id}} = (1-\mu, \mathbf{T_{id}})$.

$\mathsf{Decrypt}(\mathsf{mpk}, \mathsf{sk_{id}}, \mathsf{ct_{id}})$. On input ciphertext $(\mathbf{c}_0, \mathbf{c}'_0)$, $(\mathbf{c}_1, \mathbf{c}'_1)$, and private key $(1-\mu, \mathbf{T_{id}})$, the decryption algorithm does:

1. Compute $\mathbf{F}_{\mathsf{id},1-\mu} = [\mathbf{A}|\mathbf{A}_{1-\mu} - \mathbf{A}_{C_{\mathsf{PRF}},\mathsf{id}}]$.
2. Compute $\mathsf{m} = \mathbf{c}_{1-\mu} \oplus \mathsf{H}\left(\mathsf{Invert}(\mathbf{F}_{\mathsf{id},1-\mu}, \mathbf{T_{id}}, \mathbf{c}'_{1-\mu})\right)$.

Parameters. With $s \geq \sqrt{5} \cdot B \cdot \omega(\sqrt{\log 2m})$, we ensure that the algorithm $\mathsf{SampleLeft}$can be simulated by $\mathsf{SampleRight}$ in the security proof. We set $n \geq (n' + 2\kappa + \frac{\lambda}{\log q})\frac{\log q}{\log 2\gamma} + \frac{2\kappa}{\log q}$, $\sigma \geq 2B\beta\gamma nm$ for invoking Theorem 1. For decryption correctness, we need $\left\|\mathbf{e}_{1-\mu}^\top \cdot \mathbf{T_{id}}\right\|_\infty \leq q/4$. So we set q large enough such that $s\sigma m \leq q/4$.

If the we instantiate PRF by BMLR-PRF (Eq. 1), we can set the circuit C_{PRF} compute the function $\mathsf{MSB}(\lfloor \cdot, \cdot \rfloor)$ where the first argument of the function is, say, the first row of the identity-dependent matrix $\prod_{i=1}^{\ell} \mathbf{B}_{\mathsf{id}[i]}$ and the second argument is the secret key $\mathbf{k}$. By doing that, the PRF computation is separated into a publicly computable "heavy" part (matrix product) and a "light" part (inner-product-then-rounding). With this change, for an identity id, KeyGen and

$\mathsf{Encrypt}$ will first compute the bit string of the first row of $\prod_{i=1}^{\ell} \mathbf{B}_{\mathsf{id}[i]}$, and run Eval according to such string[2]. This makes C_{PRF} in NC^1 and we can set $d = c\log(t+\ell)$, for some constant $c > 0$, such that $q = \mathsf{poly}(\kappa)$.

4 Security

Theorem 2. *For any PPT adversary $\mathcal{A}$ against the IND-ID-CPA security of above scheme Π with advantage $\mathsf{Adv}^{\mathsf{IND\text{-}ID\text{-}CPA}}_{\mathcal{A},\Pi,(N,Q_{key},Q_{enc})}(\kappa)$, there exists PPT adversaries $\mathcal{A}_1, \mathcal{A}_2$ such that*

$$\mathsf{Adv}^{\mathsf{IND\text{-}ID\text{-}CPA}}_{\Pi,\mathcal{A},(N,Q_{key},Q_{enc})}(\kappa) \leq 3n \cdot \mathsf{Adv}^{\mathsf{LWE}_{n',q,\chi}}_{\mathcal{A}_1}(\kappa) + 2 \cdot \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{A}_2}(\kappa) + \mathsf{negl}(\kappa) \quad (2)$$

for some negligible error $\mathsf{negl}(\kappa)$.

We prove the above theorem through game-sequence technique. Let S_i denote the event that the IBE adversary $\mathcal{A}$ outputs $\mathsf{coin}' = \mathsf{coin}$ in $\mathbf{Game}_i$. We first define two simulation algorithms $\mathsf{Sim.Setup}$ and $\mathsf{Sim.KeyGen}$. which are used only for security proof. Without loss of generality, assume the adversary asks for N instances of the IBE scheme.

$\mathsf{Sim.Setup}(\mathsf{pub}, j)$. For generating parameters for j-th instance, the algorithm does the following.

1. Choose a unique system identifier $\mathsf{ID}^{(j)}$.
2. Select $\mathbf{k}^{(j)} = \mathbf{k}^{(j)}[1]...\mathbf{k}^{(j)}[t] \leftarrow \{0,1\}^t$ for PRF.
3. Select a random matrix $\mathbf{A}^{(j)} \in \mathbb{Z}_q^{n\times m}$.
4. Select $\mathbf{R}^{(j)}_{\mathbf{A}_0}, \mathbf{R}^{(j)}_{\mathbf{A}_1}, \mathbf{R}^{(j)}_{\mathbf{C}_1}, ..., \mathbf{R}^{(j)}_{\mathbf{C}_t} \leftarrow \{-1,1\}^{m\times m}$.
5. Set $\mathbf{A}^{(j)}_0 = \mathbf{A}^{(j)}\mathbf{R}^{(j)}_{\mathbf{A}_0}$, $\mathbf{A}^{(j)}_1 = \mathbf{A}^{(j)}\mathbf{R}^{(j)}_{\mathbf{A}_1} + \mathbf{G}$, and $\mathbf{C}^{(j)}_i = \mathbf{A}^{(j)}\mathbf{R}^{(j)}_{\mathbf{C}_i} + \mathbf{k}^{(j)}[i]\mathbf{G}$ for $i \in [t]$.
6. Output $\mathsf{mpk}^{(j)} = \left(\mathsf{ID}^{(j)}, \mathbf{A}^{(j)}, \mathbf{A}^{(j)}_0, \mathbf{A}^{(j)}_1, \{\mathbf{C}^{(j)}_i\}_{i\in[t]}\right)$ and $\mathsf{msk}^{(j)} = \left(\mathbf{R}^{(j)}_{\mathbf{A}_0}, \mathbf{R}^{(j)}_{\mathbf{A}_1}, \{\mathbf{R}^{(j)}_{\mathbf{C}_i}\}_{i\in[t]}\right)$.

$\mathsf{Sim.KeyGen}(\mathsf{mpk}^{(j)}, \mathsf{msk}^{(j)}, \mathsf{id})$ On input $\mathsf{id} \in \{0,1\}^{\ell}$, the algorithm does:

1. For $\mathsf{ID}^{(j)}||\mathsf{id} = \mathsf{id}[1], ..., \mathsf{id}[\ell]$, compute the $\mathbb{Z}_q^{n\times m}$-matrix

$$\begin{aligned}\mathbf{A}^{(j)}_{C_{\mathsf{PRF}},\mathsf{id}} &= \mathsf{Eval}(C_{\mathsf{PRF}}, \{\mathbf{C}^{(j)}_i\}_{i\in[t]}, \mathsf{id}[1]\mathbf{G}, ..., \mathsf{id}[\ell]\mathbf{G}) \\ &= \mathbf{A}^{(j)}\mathbf{R}^{(j)}_{C_{\mathsf{PRF}},\mathsf{id}} + \mathsf{MSB}(\mathsf{PRF}(\mathbf{k}^{(j)}, \mathsf{id}))\mathbf{G} \\ &= \mathbf{A}^{(j)}\mathbf{R}^{(j)}_{C_{\mathsf{PRF}},\mathsf{id}} + \mu\mathbf{G}\end{aligned}$$

[2] In this case we need to set parameter t to be the length of such string.

2. Set the $\mathbb{Z}_q^{n\times 2m}$-matrix

$$\begin{aligned}\mathbf{F}_{\mathsf{id},1-\mu}^{(j)} &= \left[\mathbf{A}^{(j)}|\mathbf{A}_{1-\mu}^{(j)} - \mathbf{A}_{C_{\mathsf{PRF}},\mathsf{id}}^{(j)}\right] \\ &= \left[\mathbf{A}^{(j)}|\mathbf{A}^{(j)}(\mathbf{R}_{\mathbf{A}_{1-\mu}}^{(j)} - \mathbf{R}_{C_{\mathsf{PRF}},\mathsf{id}}^{(j)}) + (1-2\mu)\mathbf{G})\right] \\ &= \left[\mathbf{A}^{(j)}|\mathbf{A}^{(j)}\mathbf{R}_{1-\mu}^{(j)} + (1-2\mu)\mathbf{G}\right]\end{aligned}$$

3. Run $\mathsf{SampleRight}(\mathbf{A}^{(j)}, \mathbf{R}_{1-\mu}^{(j)}, 1-2\mu, \mathbf{G}, s)$ to get a trapdoor $\mathbf{T}_{\mathsf{id}}^{(j)}$ for $\mathbf{F}_{\mathsf{id},1-\mu}^{(j)}$.
4. Return $\mathsf{sk}_{\mathsf{id}}^{(j)} = (1-\mu, \mathbf{T}_{\mathsf{id}})$.

The first game **Game** 0 is the same as the real IND-ID-CPA security game. **Game** 1 is the same as **Game** 1 except it runs Sim.Setup and Sim.KeyGen instead of Setup and KeyGen.

Lemma 11. **Game** *0 and* **Game** *1 are statistically indistinguishable, i.e., there exist a negligible function* $\mathsf{negl}(\lambda)$ *such that* $|\Pr[S_0] - \Pr[S_1]| \leq \mathsf{negl}(\lambda)$.

Proof. For j-th instance, the differences between **Game** 0 and **Game** 1 are:

1. In **Game** 0, $\mathbf{A}^{(j)}$ is generated by TrapGen. By Lemma 5 it has a distribution that is statistically close to uniform distribution on $\mathbb{Z}_q^{n\times m}$. On the other hand, $\mathbf{A}^{(j)}$ is sampled uniformly at random in **Game** 1.
2. By Lemma 2, matrices $\mathbf{A}_0, \mathbf{A}_0, \{\mathbf{C}_i\}_{i\in[t]}$ in **Game** 1 are statistically close to uniform distribution on $\mathbb{Z}_q^{n\times m}$. In **Game** 0 those matrices are sampled uniformly from $\mathbb{Z}_q^{n\times m}$.
3. In **Game** 0, the decryption key $\mathbf{T}_{\mathsf{id}}^{(j)}$ is sampled by SampleLeft with the trapdoor of $\mathbf{A}^{(j)}$. In **Game** 1, $\mathbf{T}_{\mathsf{id}}^{(j)}$ is sampled by SampleRight with the gadget matrix $\mathbf{G}$ and knowledge of the low-norm matrix $\mathbf{R}_{1-\mu}^{(j)}$. By Lemmas 7 and 8, for sufficiently large s (e.g., $s \geq \sqrt{5}s_1(\mathbf{R}_{1-\mu}^{(j)}) \cdot \omega(\sqrt{\log 2m})$), $\mathbf{T}_{\mathsf{id}}^{(j)}$ generated in **Game** 0 and **Game** 1 are statistically close.

We therefore conclude that **Game** 0 and **Game** 1 are statistically close up to some error $\mathsf{negl}(\lambda)$.

Game 2 is the same as **Game** 1 except that the public matrices $\{\mathbf{A}^{(j)}\}_{j\in[N]}$ for N scheme instances are generated as LWE samples. More specifically, one firstly samples $\mathbf{C} \leftarrow \mathbb{Z}_q^{n'\times n}$. For constructing $\mathbf{A}^{(j)}$, it samples $\mathbf{B}^{(j)} \leftarrow \mathbb{Z}_q^{n'\times m}$, and $\mathbf{F}^{(j)} \leftarrow \chi^{m\times n}$ and sets $\mathbf{A}^{(j)} = \mathbf{C}\cdot\mathbf{B}^{(j)} + \mathbf{F}^{(j)} \bmod q$. Here $\mathbf{C}$ serves as the secret of LWE instances for all $\mathbf{A}^{(j)}$. It is easy to see that under the LWE assumption, **Game** 2 and **Game** 3 are computationally indistinguishable. So we have the following lemma in which the factor n accounts for a n-step hybrid argument for reducing the LWE problem with matrix secret $\mathbf{C}$ to the LWE problem with single vector secret defined in Definition 3[3].

[3] Recall that the LWE problem is hard for arbitrary number of samples.

Lemma 12. $|\Pr[S_2] - \Pr[S_1]| \leq n \cdot \mathit{Adv}^{\mathit{LWE}_{n',q,\chi}}_{\mathcal{A}_1}(\lambda)$ *for some adversary* $\mathcal{A}_1$.

Game 3 is the same as **Game** 2 except that it answers the encryption query in a slightly different way. Concretely, for encryption query $(k, j, \mathsf{id}, \mathsf{m}_0, \mathsf{m}_1)$ where $k \in [Q_{\mathsf{enc}}]$, $j \in [N]$, $\mathsf{id} \in \mathcal{ID}$ and $\mathsf{m}_0, \mathsf{m}_1 \in \mathcal{M}$, encryption (of message $\mathsf{m}_{\mathsf{coin}}$) is done by normal encryption algorithm except the ciphertext component $c^{(j)}_\mu$ is chosen uniformly at random from $\{0,1\}^\lambda$, where $\mu = \mathsf{MSB}(\mathsf{PRF}(\mathbf{k}^{(j)}, \mathsf{ID}^{(j)}||\mathsf{id}))$. We have the following lemma.

Lemma 13. Game *2 and* **Game** *3 are statistically indistinguishable, i.e., there exists a negligible error* $\mathit{negl}(\kappa)$ *such that* $|\Pr[S_3] - \Pr[S_2]| \leq \mathit{negl}(\kappa)$.

Proof. First of all, we have $\mu = \mathsf{MSB}(\mathsf{PRF}(\mathbf{k}^{(j)}, \mathsf{ID}^{(j)}||\mathsf{id})) \in \{0,1\}$. By the construction of encryption algorithm, we have

$$\begin{aligned}
\mathbf{F}^{(j)}_{\mathsf{id},\mu} &= [\mathbf{A}^{(j)}|\mathbf{A}^{(j)}_\mu - \mathbf{A}^{(j)}_{C_{\mathsf{PRF}},\mathsf{id}}] \\
&= [\mathbf{A}^{(j)}|(\mathbf{A}^{(j)}\mathbf{R}^{(j)}_{\mathbf{A}_\mu} + \mu\mathbf{G}) - (\mathbf{A}^{(j)}\mathbf{R}^{(j)}_{C_{\mathsf{PRF}},\mathsf{id}} + \mathsf{MSB}(\mathsf{PRF}(\mathbf{k}^{(j)}, \mathsf{ID}^{(j)}||\mathsf{id}))\mathbf{G})] \\
&= [\mathbf{A}^{(j)}|\mathbf{A}^{(j)}(\mathbf{R}^{(j)}_{\mathbf{A}_\mu} - \mathbf{R}^{(j)}_{C_{\mathsf{PRF}},\mathsf{id}})] \\
&= [\mathbf{A}^{(j)}|\mathbf{A}^{(j)}{\mathbf{R}_\mu}^{(j)}]
\end{aligned}$$

So for the ciphertext components $(\mathbf{c}'^{(j)}_\mu, {\mathbf{c}_\mu}^{(j)})$, we have

$$\mathbf{c}'^{(j)\top}_{\mu,k} = \mathbf{x}^{(j)\top}_{\mu,k} \cdot \mathbf{F}^{(j)}_{\mathsf{id},\mu} + \mathbf{e}^{(j)\top}_{\mu,k} \quad ; \quad \mathbf{c}^{(j)}_{\mu,k} = \mathsf{m}_{\mathsf{coin}} \oplus \mathsf{H}(\mathbf{x}^{(j)}_{\mu,k})$$

where $\mathbf{x}^{(j)}_{\mu,k}$, $e^{(j)}_{\mu,k}$ are chosen randomly and freshly for each ciphertext query with index k.[4] Recall $\mathbf{A}^{(j)} = \mathbf{C} \cdot \mathbf{B}^{(j)} + \mathbf{F}^{(j)}$ where $\mathbf{B}^{(j)} \in \mathbb{Z}_q^{n' \times m}$ is randomly chosen and $m = 2n \log q + \omega(\log q)$. By Lemma 2, $\mathbf{B}^{(j)}\mathbf{R}^{(j)}_\mu$ is statistically close to uniform (given $\mathbf{F}^{(j)}\mathbf{R}^{(j)}_\mu$) by itself, as required by Theorem 1. Since here we consider the left entropy of randomly and independently chosen $\mathbf{x}^{(j)}_{\mu,k}$, we can still apply Theorem 1 even though $\mathbf{B}^{(j)}\mathbf{R}^{(j)}_\mu$ is not statistically uniform given $\mathbf{F}^{(j)}_{\mathsf{id}',\mu}$ from another encryption query with $\mathsf{id}' \neq \mathsf{id}$. By Theorem 1 we get

$$\begin{aligned}
H_\infty\left({\mathbf{x}_{\mu,k}}^{(j)}|\mathbf{c}'^{(j)}_{\mu,k}\right) &\geq H_\infty\left(\mathbf{x}^{(j)}_{\mu,k}\right) - (n' + 2\kappa)\log q \\
&\geq n\log(2\gamma) - (n' + 2\kappa)\log q \\
&\geq \lambda + 2\kappa
\end{aligned}$$

By Lemma 1, we have

$$\Delta\left((\mathsf{H}, \mathsf{H}(\mathbf{x}^{(j)}_{\mu,k})), (\mathsf{H}, \rho^{(j)}_k)\right) \leq 2^{-\kappa} = \mathsf{negl}(\kappa)$$

for uniformly random string $\rho^{(j)}_k \leftarrow \{0,1\}^\lambda$. This makes $\mathbf{c}^{(j)}_{\mu,k}$ uniformly random and independent of $\mathsf{m}_{\mathsf{coin}}$.

[4] This is why our scheme achieves strong/full adaptive MIMC security.

Game 4 is the same as **Game** 3 except that it uses $\mathsf{Sim.Setup}$ to generate the public parameters. In particular, $\mathbf{A}^{(j)}$ is sampled uniform at random. Looking ahead, this step allows us to run Setup (instead of $\mathsf{Sim.Setup}$) in the next game where we are able to have trapdoor for the matrix $\mathbf{A}^{(j)}$. A straightforward reduction gives us the following lemma.

Lemma 14. $|\Pr[S_4] - \Pr[S_3]| \leq n \cdot \mathsf{Adv}^{\mathsf{LWE}_{n',q,\chi}}_{\mathcal{A}_1}(\kappa)$ *for some adversary* $\mathcal{A}_1$.

Game 5 is the same as **Game** 4 except that it runs algorithms Setup and KeyGen instead of the simulation algorithms. Similar to Lemma 11, we have

Lemma 15. Game *4 and* **Game** *5 are statistically indistinguishable, i.e.* $|\Pr[S_5] - \Pr[S_4]| \leq \mathsf{negl}(\kappa)$ *for some negligible function* $\mathsf{negl}(\kappa)$.

Game 6 is the same as **Game** 5 except that the simulator samples the bit value μ uniformly instead of computing it by PRF as in **Game** 5. The simulator also keeps the record of tuples (j, id, μ). For a private key generation query or encryption query on instance j and identity id that has been made before, the simulator simply finds the recorded μ and uses it for further operations. We prove the following lemma.

Lemma 16. $|\Pr[S_6] - \Pr[S_5]| \leq \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{A}_2}(\kappa)$ *for some adversary* $\mathcal{A}_2$ *against* PRF.

Proof. We build a simulator $\mathcal{A}_2$ who uses a PRF challenger to simulate **Game** 5 or **Game** 6. $\mathcal{A}_2$ flips a fair coin $\mathsf{coin} \in \{0,1\}$ and follows $\mathsf{Para}(1^\kappa)$ to generate all the parameters of pub except the almost key-homomorphic PRF PRF. Instead, $\mathcal{A}_2$ receives PRF from its challenger.

$\mathcal{A}_2$ chooses N random PRF keys $\{\tilde{\mathbf{k}}^{(j)}\}_{j\in[N]}$. Then it runs Setup to generate $\{\mathsf{mpk}^{(j)}, \mathsf{msk}^{(j)}\}_{j\in[N]}$ except the PRF keys. Notice that $\mathsf{mpk}^{(j)}$ has exactly the same distribution as in the real scheme. $\mathcal{A}_2$ answers the following two types of query.

1. For a key generation query (j, id), $\mathcal{A}_2$ first sends $\mathsf{ID}^{(j)}||\mathsf{id}$ to its challenger and receives back y. It sets $\mu = \mathsf{MSB}\left(y + \mathsf{PRF}(\tilde{\mathbf{k}}^{(j)}, \mathsf{ID}^{(j)}||\mathsf{id})\right)$ and runs steps 2 to step 5 of KeyGen to generates the private identity key.
2. For an encryption query $(k, j, \mathsf{id}, \mathsf{m}_0, \mathsf{m}_1)$, $\mathcal{A}_2$ first sends $\mathsf{ID}^{(j)}||\mathsf{id}$ to its challenger, receives back y, and sets $\mu = \mathsf{MSB}(y + \mathsf{PRF}(\tilde{\mathbf{k}}^{(j)}, \mathsf{ID}^{(j)}||\mathsf{id}))$. It then runs $\mathsf{Encrypt}$ to generate ciphertext on message $\mathsf{m}_{\mathsf{coin}}$ based on the bit value μ, except it samples the component $c^{(j)}_{\mu,k}$ randomly.

If $y = \mathsf{PRF}(\mathbf{k}^*, \mathsf{ID}^{(j)}||\mathsf{id})$ for some key $\mathbf{k}^*$, i.e., $\mathcal{A}_2$ interacts with PRF, we have

$$\begin{aligned}\mu &= \mathsf{MSB}(y + \mathsf{PRF}(\tilde{\mathbf{k}}^{(j)}, \mathsf{ID}^{(j)}||\mathsf{id})) \\ &= \mathsf{MSB}(\mathsf{PRF}(\mathbf{k}^*, \mathsf{ID}^{(j)}||\mathsf{id}) + \mathsf{PRF}(\tilde{\mathbf{k}}^{(j)}, \mathsf{ID}^{(j)}||\mathsf{id})) \\ &= \mathsf{MSB}(\mathsf{PRF}(\mathbf{k}^* \boxplus \tilde{\mathbf{k}}^{(j)}, \mathsf{ID}^{(j)}||\mathsf{id}))\end{aligned}$$

This shows that $\mathcal{A}_2$ simulates **Game** 5 with random PRF key $\mathbf{k}^{(j)} = \mathbf{k}^* + \tilde{\mathbf{k}}^{(j)}$. On the other hand, if $y = F(\mathsf{ID}^{(j)}||\mathsf{id})$ for some random function $F : \{0,1\}^\ell \to \{0,1\}^r$, as $F(\cdot)$ is never takes the same input, μ is uniformly random from the adversary's view. In this case, $\mathcal{A}_2$ simulates **Game** 6. Therefore we have $\Pr[S_6] - \Pr[S_5] \leq \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{A}_2}(\kappa)$.

Let $(\mathbf{c}^{(j)}_{0,k}, \mathbf{c'}^{(j)}_{0,k}, \mathbf{c}^{(j)}_{1,k}, \mathbf{c'}^{(j)}_{1,k})$ be the challenge ciphertext generated for answering the k-th encryption query $(k, j, \mathsf{id}, \mathsf{m}_0, \mathsf{m}_1)$. Recall that in **Game** 6, depending on the bit value $\mu = \mathsf{MSB}(\mathsf{PRF}(\mathbf{k}^{(j)}, \mathsf{id}))$, $\mathbf{c}_\mu{}^{(j)}$ is chosen randomly. **Game** 7 is the same as **Game** 6 except that it chooses $\mathbf{c}_{1-\mu,k}{}^{(j)}$ randomly and computes other components honestly. Since μ is random, we have the following lemma.

Lemma 17. **Game** *6 and* **Game** *7 are identical, i.e.,* $\Pr[S_6] = \Pr[S_7]$.

Game 8 is the same as **Game** 7 except that for encryption and key generation queries on j-th instance and identity id, the bit value μ is computed as $\mu = \mathsf{MSB}(\mathsf{PRF}(\mathbf{k}^{(j)}, \mathsf{ID}^{(j)}||\mathsf{id}))$. Similar to Lemma 16, we have the following lemma for which we omit the proof as it is identical to the proof of Lemma 16.

Lemma 18. $|\Pr[S_7] - \Pr[S_8]| \leq \mathit{Adv}^{\mathit{PRF}}_{\mathcal{A}_2}(\kappa)$ *for some adversary* $\mathcal{A}_2$ *against PRF.*

Game 9 is the same as **Game** 8 except that the simulation algorithms Sim,Setup and Sim.KeyGen are invoked instead of Setup and KeyGen. Notice that this difference is exactly the difference between **Game** 0 and **Game** 1. So we have the following lemma which can be proved using the proof of Lemma 11.

Lemma 19. **Game** *8 and* **Game** *9 are statistically indistinguishable, i.e., there exist a negligible function* $\mathit{negl}(\kappa)$ *such that* $|\Pr[S_9] - \Pr[S_8]| \leq \mathit{negl}(\kappa)$.

In the next game **Game** 10, instead of sampling the public matrices $\{\mathbf{A}^{(j)}\}_{j\in[N]}$ for N instances randomly, we again generate them by LWE samples as in **Game** 2, i.e., $\mathbf{A}^{(j)} = \mathbf{C} \cdot \mathbf{B}^{(j)} + \mathbf{F}^{(j)} \bmod q$. This change is not noticeable for efficient adversary under LWE assumption which can be stated by the lemma below.

Lemma 20. $|\Pr[S_{10}] - \Pr[S_9]| \leq n \cdot \mathit{Adv}^{\mathit{LWE}_{n',q,\chi}}_{\mathcal{A}_1}(\kappa)$ *for some adversary* $\mathcal{A}_1$.

Game 11 is the same as **Game** 10 except that for any encryption query $(k, j, \mathsf{id}, \mathsf{m}_0, \mathsf{m}_1)$, the ciphertext component $\mathbf{c}^{(j)}_{\mu,k}$ are chosen randomly, where $\mu = \mathsf{PRF}(\mathbf{k}^{(j)}, \mathsf{ID}^{(j)}||\mathsf{id})$. Notice that we have already switched the ciphertext component $\mathbf{c}^{(j)}_{1-\mu,k}$ to random since **Game** 7. So in **Game** 11, both $\mathbf{c}_{0,k}{}^{(j)}$ and $\mathbf{c}_{1,k}{}^{(j)}$ (which were used to mask the message $\mathsf{m}_{\mathsf{coin}}$) are random, meaning that the challenge ciphertexts replied to encryption queries are random and independent of the messages chosen by the adversary. So the adversary has no advantage in wining **Game** 11. The proof of the following lemma is omitted as it is the same as the proof of Lemma 13.

Lemma 21. Game *10 is statistically close to* **Game** *11, and in* **Game** *11, no adversary has any advantage in guessing the bit* coin*, i.e.,* $|\Pr[S_{11}] - \Pr[S_{10}]| \leq \mathsf{negl}(\kappa)$ *for some statistically error* $negl(\kappa)$ *and* $\Pr[S_{11}] = 1/2$.

To sum up, we have:

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{IND\text{-}ID\text{-}CPA}}_{\Pi,\mathcal{A},(N,Q_{\mathsf{key}},Q_{\mathsf{enc}})}(\kappa) &= |\Pr[S_1] - 1/2| \\
&= |\Pr[S_1] - \Pr[S_{11}]| \\
&\leq \sum\nolimits_{i=0}^{10} |\Pr[S_i] - \Pr[S_{i+1}]| \\
&= 3n \cdot \mathsf{Adv}^{\mathsf{LWE}_{n',q,\chi}}_{\mathcal{A}_1} + 2 \cdot \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{A}_2}(\kappa) + \mathsf{negl}(\kappa)
\end{aligned}$$

for some function $\mathsf{negl}(\kappa)$ which stands for the negligible statistical error in the reduction. The security loss is independent of the number of instances N, the number of encryption queries Q_{enc} and the number of key generation queries Q_{key}.

5 Discussion and Conclusion

For generality, we reduce the security of the IBE scheme to the LWE problem $\mathsf{LWE}_{n',q,\chi}$ and the security of the PRF as shown by Theorem 2. To make the whole IBE scheme (almost) tightly secure, we need (almost) tightly secure PRFs. The instantiation of PRF also affects the LWE problem $\mathsf{LWE}_{n',q,\chi}$ quantitatively by the depth d of the circuit C_{PRF}. For example, employing an almost tightly secure (based on the LWE problem) BLMR-PRF [10] allows us to use a polynomial modulo q. Meanwhile, the computational assumption we make for the PRF affects the final assumption that we need to make for the IBE scheme. The (almost) tight security proof of the BLMR-PRF requires an LWE assumption with modulus-to-noise ratio $n^{\Omega(\ell)}$ (ℓ is the PRF input length) which is quantitatively stronger than the LWE problem $\mathsf{LWE}_{n',q,\chi}$ we use for the LWE lossy mode. This means the IBE scheme needs a strong LWE assumption on which the BLMR-PRF is based. However any future improvement in (lattice-based) key-homomorphic PRFs will directly improve the efficiency and security of our scheme without weakening the underlying assumption.

Under a suitable BLMR-PRF instantiation, our IBE scheme, based on a strong LWE assumption (sub-exponential modulus-to-noise ratio), achieves almost tight security in the strong MIMC setting. Under the same assumption, the Boyen-Li IBE scheme from [11] (using almost tightly secure PRFs from [5,10]) only had an almost tight security reduction in the SISC setting. The (strong) LWE assumption that we use is believed to be hard and has been widely used in other contexts, including fully-homomorphic encryption [14], attribute-based/predicate encryption [9,21,22] and lattice-based constrained PRFs [13,15]. How to obtain an (almost) tightly secure IBE scheme in the MIMC setting was not known before, even with such a strong LWE assumption. By applying the standard BCHK transformation [8] with tightly secure

one-time signature schemes (e.g., [7]), our IBE scheme leads to the first almost tightly CCA2 secure public-key encryption scheme from lattices in the multi-instance and multi-ciphertext setting [23].

Our work motivates two future directions: to improve efficiency and key sizes; and to design tightly secure key-homomorphic PRFs from weaker assumptions.

References

1. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_28
2. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: STOC, pp. 99–108 (1996)
3. Alwen, J., Krenn, S., Pietrzak, K., Wichs, D.: Learning with rounding, revisited. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8042, pp. 57–74. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40041-4_4
4. Attrapadung, N., Hanaoka, G., Yamada, S.: A framework for identity-based encryption with almost tight security. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9452, pp. 521–549. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48797-6_22
5. Banerjee, A., Peikert, C., Rosen, A.: Pseudorandom functions and lattices. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 719–737. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_42
6. Bellare, M., Kiltz, E., Peikert, C., Waters, B.: Identity-based (lossy) trapdoor functions and applications. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 228–245. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_15
7. Blazy, O., Kakvi, S.A., Kiltz, E., Pan, J.: Tightly-secure signatures from chameleon hash functions. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 256–279. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46447-2_12
8. Boneh, D., Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. SIAM J. Comput. **36**(5), 1301–1328 (2006)
9. Boneh, D., Gentry, C., Gorbunov, S., Halevi, S., Nikolaenko, V., Segev, G., Vaikuntanathan, V., Vinayagamurthy, D.: Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 533–556. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55220-5_30
10. Boneh, D., Lewi, K., Montgomery, H., Raghunathan, A.: Key homomorphic PRFs and their applications. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8042, pp. 410–428. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40041-4_23
11. Boyen, X., Li, Q.: Towards tightly secure lattice short signature and id-based encryption. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10032, pp. 404–434. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53890-6_14
12. Boyen, X., Li, Q.: All-but-many lossy trapdoor functions from lattices and applications. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10403, pp. 298–331. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63697-9_11

13. Brakerski, Z., Tsabary, R., Vaikuntanathan, V., Wee, H.: Private constrained PRFs (and more) from LWE. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017. LNCS, vol. 10677, pp. 264–302. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70500-2_10
14. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. In: FOCS, pp. 97–106 (2011)
15. Brakerski, Z., Vaikuntanathan, V.: Constrained key-homomorphic PRFs from standard lattice assumptions. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015. LNCS, vol. 9015, pp. 1–30. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46497-7_1
16. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. J. Cryptol. **25**(4), 601–639 (2012)
17. Chen, J., Gong, J., Weng, J.: Tightly secure IBE under constant-size master public key. In: Fehr, S. (ed.) PKC 2017. LNCS, vol. 10174, pp. 207–231. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54365-8_9
18. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC, pp. 197–206 (2008)
19. Gong, J., Chen, J., Dong, X., Cao, Z., Tang, S.: Extended nested dual system groups, revisited. In: Cheng, C.-M., Chung, K.-M., Persiano, G., Yang, B.-Y. (eds.) PKC 2016. LNCS, vol. 9614, pp. 133–163. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49384-7_6
20. Gong, J., Dong, X., Chen, J., Cao, Z.: Efficient IBE with tight reduction to standard assumption in the multi-challenge setting. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10032, pp. 624–654. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53890-6_21
21. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Attribute-based encryption for circuits. In: STOC, pp. 545–554 (2013)
22. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Predicate encryption for circuits from LWE. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9216, pp. 503–523. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48000-7_25
23. Hofheinz, D., Jager, T.: Tightly secure signatures and public-key encryption. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 590–607. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_35
24. Hofheinz, D., Koch, J., Striecks, C.: Identity-based encryption with (almost) tight security in the multi-instance, multi-ciphertext setting. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 799–822. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46447-2_36
25. Libert, B., Sakzad, A., Stehlé, D., Steinfeld, R.: All-but-many lossy trapdoor functions and selective opening chosen-ciphertext security from LWE. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10403, pp. 332–364. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63697-9_12
26. Micciancio, D., Peikert, C.: Trapdoors for lattices: simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_41
27. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. SIAM J. Comput. **40**(6), 1803–1844 (2011)
28. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC, pp. 84–93 (2005)