Compact Partitioning from Truncation Collision-Resistant Hash: Application to the Lattice-based IBEs

Parhat Abla

College of Mathematics and Computer Science, Shantou University
parhat@stu.edu.cn.

Abstract. A lattice-based adaptively secure compact IBE scheme can be constructed in the random oracle model. However, not only is it challenging to construct a lattice-based compact IBE in the standard model, but the compact construction from truncation collision-resistant hash functions (TCRHF) introduced by Jager and Kurek (ASIACRYPT 2018) still fails to construct a compact IBE.
To fill this gap, we construct an adaptively secure, compact IBE scheme from the lattice assumption and the TCRHF. Compared with the previous IBE scheme of Jager et al. (PKC 2021) from a variant TCRHF, in which the master public key includes $\omega(\log(\lambda))$ matrices (or ring vectors if considering in the ring setting), our scheme only needs a constant number of ring vectors in the master public key. To achieve this compact IBE, we first design a compact partition function based on the TCRHF and an equality-testing function. We believe our newly designed partition function can be applied to the design of other compact cryptographic schemes based on TCRHF.

Keywords: Lattices · IBE · Ring LWE

1 Introduction

Identity-Based Encryption (IBE) extends the concept of public-key encryption by allowing a user's public key to be derived directly from a unique identifier, such as an email address or a unique name. Encrypted messages can be sent to a user using only their identity as the public key, while the corresponding decryption key is obtained from a trusted authority known as the Private Key Generator (PKG). The primary advantage of IBE is the elimination of traditional public key infrastructure (PKI), which streamlines key management and distribution. An IBE scheme is called adaptively secure if, even when an attacker can request private keys for arbitrary identities (except the target) and adapt their choices based on information learned during the attack, they still cannot distinguish between encryptions under the target identity. This is a stronger notion than selective security, where the adversary must declare the target identity before seeing the system's public parameters. Many influential adaptively

secure IBE constructions [5, 23, 24] are based on classical number-theoretic assumptions; however, these assumptions are vulnerable to quantum attacks [22]. This motivates the study of post-quantum IBE, and lattice-based cryptography is among the most compelling candidates due to its worst-case hardness and its rich toolkit for advanced primitives. Therefore, this paper focuses on the constructions of adaptively secure lattice-based IBE schemes.

The first lattice-based IBE scheme presented in [8] uses the pre-image sampling technique, but it is only proven in the random-oracle model. The works of [3, 7] construct the first lattice-based IBE in the standard model, but their master public key (mpk) consists of $O(\ell)$ random matrices. Therefore, the construction is not compact[1].

On one hand, a series of works [1, 2, 4, 9, 13–16, 25–27] have applied partitioning techniques to reduce the public parameter size of lattice-based IBE schemes. However, none of these constructions achieves truly *compact* IBE. Among them, the asymptotically most compact scheme is presented in [2], where the IBE scheme still requires at least $\omega(1)$ ring vectors in the master public key, which remains superconstant. We refer to the IBE scheme of [2] as achieving almost compactness.On the other hand, [12] constructs a compact IBE from the LWE assumption and a variant of the truncation collision-resistant hash function (TCRHF)[2] introduced in [11]. However, their construction needs $\omega(\log(\lambda))$ matrices in the mpk. This result performs less favorably compared to the construction in [2]. Therefore, a natural question that comes to mind is:

Can we construct a *compact* adaptively secure IBE scheme from standard lattice assumptions and TCRHF ?

The works [6,17] focus on constructing (almost) tightly secure IBE from lattices. However, their schemes are no more compact than the first adaptively secure IBE schemes in [3,7]. We therefore aim to construct an adaptively secure, short, and compact lattice-based IBE scheme.

[1] We call an adaptively secure IBE scheme (almost) compact if its mpk size is $(\omega(1))$ constant times the size of that of a selectively secure IBE scheme. In the lattice setting, it is folklore to call an adaptively secure IBE scheme compact if it contains a constant number of matrices in $\mathbb{Z}_q^{n\times m}$ (or ring vectors in $\mathbb{Z}_q[x]/(x^n+1)$) for a scheme modulus q of polynomial size in the security parameter λ, and $n = \Theta(\lambda), m = 2n\cdot\log(q)$. Therefore, the asymptotic number of matrix (or ring vectors) can directly show the asymptotic bit size of the mpk.

[2] A TCRHF hashes an input and uses only a truncated output. The truncation collision resistance property requires that no algorithm can find collisions significantly faster than the standard birthday collision algorithm, even when only short prefixes of hash values are considered [11]. The work of [12] requires the hash function to satisfy the weak n'-near-collision resistance, in which the adversary's goal is to find the collisions at n' positions that were claimed by the adversary at the beginning of the security game [12]. If these positions are selected at the first n' places, weak n'-near-collision resistance can be reduced to the truncation collision resistance.

1.1 Our Contributions

We answer the above question positively by constructing a lattice-based, partition-based IBE scheme with short master public keys using TCRHF. Following the research line of partition-based constructions, we obtain an adaptively secure *compact* IBE scheme from ring learning with errors (RLWE, see Definition 2.11) problems and TCRHF. Namely, the mpk in our IBE scheme only includes a constant number of ring vectors, which is comparable with the selectively secure IBE construction of [3]. Furthermore, our IBE scheme shares relatively small parameters (asymptotically) than the previous almost compact IBE construction of [2]. To achieve this compact IBE scheme, we combine the more compact homomorphic equality test function given by Abla et al. [2] and the TCRHF introduced by [11]. The detailed construction ideal will be presented in the next section.

In Table 1, we summarize our results and compare them with the previous partition-based *typical* constructions. To be fair, we considered the ring variant of the IBE schemes based on the plain LWE problem. Due to the cost of large public key sizes (need for super-polynomially large modulus and a large number of matrices in the master public keys), we didn't list those (almost) tight constructions of lattice-based IBEs in the standard model [6,17]. Note that the IBE scheme of Zhang et al. [27] is Q-bounded; thus, we didn't list it in the table.

Table 1. Comparison with Lattice Based IBE Schemes in the Ring Setting.

Scheme	# of ring vectors in mpk	# of ring vectors in $\mathsf{ct}/\mathsf{sk}_{\mathsf{id}}$	R-LWE Param $\frac{1}{\alpha}$
[3]	$O(\lambda)$	$O(1)$	$\tilde{O}(n^{3.5})$
[16]	$O(\lambda^{1/\mu})^{\dagger}$	$O(1)$	$O(n^{0.5+2\mu})$
[26] I + [15]	$\omega(\log^2(\lambda))$	$O(1)$	$\tilde{O}(n^{5.5})$
[26] II	$\omega(\log(\lambda))$	$O(1)$	$\mathsf{poly}(n)^{*}$
[12]	$O(\log(\lambda))$	$O(1)$	$\tilde{O}(n^{6})$
[1]	$\omega(\log(n)/\log\log(n))$	$O(1)$	$\tilde{O}(n^{8.5})$
[2] A	$\omega(\log(\lambda))$	$O(1)$	$O(n^{4.5+\frac{4}{\kappa}})^{\dagger\dagger}$
[2] B	$\omega(1)$	$O(1)$	$\tilde{O}(n^{7.5+\frac{4}{\kappa}})$
Ours	$O(1)$	$O(1)$	$\tilde{O}(n^{5.5})^{\ddagger}$

Notations: $\mathsf{mpk}, \mathsf{ct}, \mathsf{sk}_{\mathsf{id}}$ denote the master public key, ciphertext, and secret key of identity id. α, λ and n denotes R-DLWE parameter, security parameter and the ring dimension.

($\ddagger$) This value can be reduced to $\tilde{\mathsf{O}}(\mathsf{n}^{4.5})$ if we consider ours IBE scheme in the *crs* model as in [2] A.

$*$ $\mathsf{poly}(n)$ denotes some fixed but large polynomial. It is hard to determine an explicit bound for comparison due to the implicit construction of the work.

$\dagger$ $\mu \in \mathbb{N}$ is a constant that can be chosen arbitrarily. Since the reduction cost is exponential in μ, this value is typically set very small (e.g., $\mu = 2$ or 3).

$\dagger\dagger$ $\kappa \geq 1$ can be any constant that satisfies $n^{\frac{1}{\kappa}} > 3 + \kappa$, e.g., 2 or 4, depending on how we set parameters of the underlying error correcting code.

1.2 Overview of Our Techniques

In this section, we provide an overview of our high-level construction ideas.

Previous design idea. Recall that, in the adaptive security game of the IBE scheme, the adversary is allowed to make key extraction requests to KeyGen for any non-challenge identity id at any stage of the security game, so the simulating algorithm should be able to respond with a corresponding legal secret key $\mathsf{sk}_{\mathsf{id}}$. Meanwhile, to reduce the security of the IBE scheme to the associated RLWE problem, the simulating algorithm must embed the RLWE challenges into the challenge ciphertexts of the IBE scheme. In general, the identity space is divided into two parts: for one part, the simulation algorithm can generate secret keys; for the other, it can generate challenge-ciphertexts. If a keyed function $F_K(\cdot)$ can split the identity space as desired, then the simulating algorithm can use this function to do the reduction. We call the function $F_K(\cdot)$ a partition function. More compact (requiring fewer inputs) partition functions result in more compact IBE schemes. Partition-based IBE schemes, as their name suggests, make use of the above intuitive observation. Therefore, the most challenging part of this approach is to design a more compact partition function.

While Considering the above intuition in lattice settings as in [3], the simulating algorithm can generate the secret key $\mathsf{sk}_{\mathsf{id}}$ via the efficient trapdoor of [19] if the function value $F_K(\mathsf{id}) = 1$, and can embed the RLWE challenges into the challenge ciphertext of id^* if $F_K(\mathsf{id}^*) = 0$. The work of [26] generalized the above property of $F_K(\mathsf{id})$ as follows:

$$F_K\left(\mathsf{id}^{(1)}\right) = 1 \wedge \cdots \wedge F_K\left(\mathsf{id}^{Q)}\right) = 1 \wedge F_K\left(\mathsf{id}^*\right) = 0,$$

where $\mathsf{id}^{(1)} \cdots \mathsf{id}^{(Q)}$ are identities that the adversary asked for key extraction, and id^* is the challenge identity. As shown in [26], a lattice-based IBE scheme can be constructed by (1) designing a partition function with the above property[3], and (2) designing efficient homomorphic evaluation algorithms for the partition function. Following this idea, the follow-up work [2] showed that pairwise independent hash functions suffice for partitioning, and constructed a partition function from error correcting code (ECC) by repeating the basic hash function $H_{\alpha,\beta}(\mathsf{id}) := \mathsf{ECC}(\mathsf{id})[\alpha] + \beta \in \mathbb{Z}_p$ for $t = \omega(1)$ times[4] .Then, we considered the whole design in the ring setting and presented an efficient homomorphic evaluation of the hash function. Note that the hash function above includes a partitioning key in its definition. Below, we show our compact partition-function design with keyless TCRHF.

Our Techniques. We observe that the function value $F_K(\mathsf{id}) = 1$ for a non-challenge identity id is unnecessary for a lattice-based IBE. Instead, we need

[3] We further need the partition probability to be noticeable, and can be covered in a sufficiently small interval.

[4] The error correcting code is defined as $\mathsf{ECC} : \{0,1\}^\ell \to \mathbb{Z}_p^{\mathsf{L}}$ for some prime p and integer L, $\mathsf{ECC}(\mathsf{id})[\alpha]$ denotes the α-th element of the vector $\mathsf{ECC}(\mathsf{id})$ for $\mathsf{id} \in \{0,1\}^\ell$.

$F_K(\mathsf{id})$ to be invertible in the corresponding ring R_q. Yet the value $F_K(\mathsf{id}^*) = 0$ is required. In addition, we observe that the partition function can be seen as the composition of a pairwise independent key-less hash function and another keyed function (e.g., $F_K(\mathsf{id}) := f_K(\mathsf{Hash}(\mathsf{id}))$), where Hash is a key-less public function. Using this important observation, we can combine the TCRHF from [12] with the equality test function from [2] as follows: Firstly, for a key $K = (t^*, i^*) \in \{0,1\}^d \times [d]$ for some $d \geq \omega(\log(n))$, we define the function $f_K : \{0,1\}^d \to R_q$ as

$$f_K(y) := y_{\leq i^*}(x) - t^*_{\leq i^*}(x),$$

where $y_{\leq i^*}(x)$ and $t^*_{\leq i^*}(x)$ are the ring elements with the coefficients from the corresponding i^*-prefixes of the strings y and t^*. Then, for a key $K = (t^*, i^*) \in \{0,1\}^d \times [d]$, our final partition function, $F_K(\cdot)$, is defined as

$$F_K(\mathsf{id}) := f_K(\mathsf{Hash}(\mathsf{id})) = \mathsf{Hash}(\mathsf{id})_{\leq i^*}(x) - t^*_{\leq i^*}(x) \in R_q,$$

where $\mathsf{Hash} : \{0,1\}^\lambda \to \{0,1\}^d$ is a truncation collision-resistant hash function [5], and $t^*_{\leq i^*}(x)$, $\mathsf{Hash}(\mathsf{id})_{\leq i^*}(x)$ are the binary coefficient ring elements, whose entries are from the corresponding bits of the bit strings $\mathsf{Hash}(\mathsf{id})$ and t^*. Note that the function $F_K(\mathsf{id})$ is trinary polynomial, and thus for some special quotient ring [6], it is invertible if it is non-zero (see Lemma 2.8). Namely, $F_K(\mathsf{id})$ is a zero polynomial if it is non-invertible. Furthermore, the partitioning property can be shown from the truncation collision resistance property of the hash function Hash and the design structure of the function $F_K(\cdot)$. Another desired property of the partition function is efficient homomorphic evaluation with polynomial modulus. Below, we show this property of the above function $F_K(\cdot)$.

To homomorphic evaluate the function F_K , we rewrite it as follows:

$$F_K(\mathsf{id}) = \sum_{i=1}^{d} \mathsf{Eqtest}_i(i^*) \cdot \mathsf{Hash}(\mathsf{id})_{\leq i} - t^*_{\leq i^*},$$

where $\mathsf{Eqtest}_i(i^*)$ is the equality test function that outputs 1 if $i = i^*$, and outputs 0 otherwise. Therefore, given the GSW-type encodings, as in [2], of i^* and $t^*_{\leq i^*}$, then the homomorphic evaluation of F_K can be accomplished by the linear combination of the homomorphic evaluation of the equality test function $\mathsf{Eqtest}_i(\cdot)$ for $i \in [d]$. Thus, the above evaluation process is compact and efficient if the underlying equality test function is efficient and compact. Fortunately, the work of [2] presented an efficient and compact homomorphic evaluation of the equality test function Eqtest_i for the index $i \in [n]$. From the previous discussion, the partition property holds for $d \geq \omega(\log(n))$. Therefore, the homomorphic evaluation process presented above is compact and efficient.

Furthermore, due to the compactness of our partition function, our IBE scheme using the partition technique is also compact. We also believe that our newly designed partition function may find applications in designing other compact lattice-based cryptographic schemes, such as signatures, ABE, and others.

[5] More details can be found in the following sections or the work [12].

[6] Let $R_q = \mathbb{Z}[x]/(x^n + 1)$ for n a power of 2, and q be a prime such that $q = 3 \mod 8$, then the polynomial $a \in R_q$ with $\|\mathsf{Coeffs}(a)\| \leq \sqrt{q}$ is invertible in R_q

2 Preliminaries

Notations. We use $[k]$ to denote the index set $\{1, 2, \cdots, k\}$. We use lower-case bold letters to denote the vectors (e.g., $\mathbf{a}$), and use upper-case bold letters to denote the matrices (e.g., $\mathbf{A}$). For a probability distribution χ, we use $x \leftarrow \chi$ to denote that x is sampled from the distribution χ. For a set S, $x \xleftarrow{\$} S$ denotes that x is uniformly sampled from the set S. The 2-norm of a vector $\mathbf{a}$ is denoted by $\|\mathbf{a}\|$. For a square matrix $\mathbf{T}$, we use $\|\mathbf{T}\|_{\mathsf{GS}}$ to denote the longest column of Gram-Schmidt orthogonalization of $\mathbf{T}$ and use $s_1(\mathbf{T})$ to denote the largest spectral norm of $\mathbf{T}$. The function $\mathsf{negl}(\cdot)$ denotes the negligible function, that is for any constant c, there is an λ_0 such that for all $\lambda > \lambda_0$ we have $\mathsf{negl}(\lambda) < \frac{1}{\lambda^c}$. We define the statistical distance between two random variables $\mathcal{X}$ and $\mathcal{Y}$ over the support Ω as $\Delta(X, Y) := \sum_{s \in \Omega} |\mathsf{Pr}[X = s] - \mathsf{Pr}[Y = s]|$. For two probability distributions $\mathcal{P}$ and $\mathcal{Q}$, we say $\mathcal{P}$ is statistically close to $\mathcal{Q}$ if $\Delta(\mathcal{P}, \mathcal{Q}) \leq \mathsf{negl}(\lambda)$.

2.1 Identity Based Encryption

Definition 2.1 *An* IBE *scheme consists of algorithm tuples* $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *as follows:*

$\mathsf{Setup}(\lambda) \to (\mathsf{mpk}, \mathsf{msk})$: *on input the security parameter* λ*, it outputs the master public key* mpk *and master secret key* msk.

$\mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, \mathsf{id}) \to \mathsf{sk}_{\mathsf{id}}$: *on input the master key pair* $(\mathsf{mpk}, \mathsf{msk})$ *and an identity* id*, it outputs the secret key* $\mathsf{sk}_{\mathsf{id}}$ *for the corresponding identity* id.

$\mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, \mu) \to \mathsf{ct}$: *on input the master public key* mpk*, an identity* id *and a message* μ*, it outputs the ciphertext* ct *of the message* μ *under the identity* id.

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{ct}, \mathsf{sk}_{\mathsf{id}}) \to \mu/\perp$: *on input the master public key* mpk*, ciphertext* ct *and the secret key* $\mathsf{sk}_{\mathsf{id}}$ *of identity* id*, it outputs a message* μ *if* ct *is the encryption of* μ *under the identity* id*, and outputs* $\perp$ *otherwise.*

Correctness. We say an IBE scheme Π_{IBE} is δ-correct if for the security parameter λ, identity $\mathsf{id} \in \mathsf{ID}$ and any message μ, the following holds

$$\mathsf{Pr}\left[\mu' \neq \mu \;\middle|\; \begin{array}{rl} (\mathsf{mpk}, \mathsf{msk}) & \leftarrow \mathsf{Setup}(\lambda); \\ \mathsf{sk}_{\mathsf{id}} & \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, \mathsf{id}); \\ \mathsf{ct} & \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, \mu); \\ \mu' & \leftarrow \mathsf{Dec}(\mathsf{mpk}, \mathsf{ct}, \mathsf{sk}_{\mathsf{id}}) \end{array}\right] < \delta,$$

where the probability is taken over the randomness of the algorithms $\mathsf{Setup}, \mathsf{KeyGen}$ and the encryption algorithm Enc. If the value of δ is negligible in the security parameter λ, then we simply call the IBE scheme correct.

Adaptive Security. The adaptive security of an IBE scheme is described by the following experiment $\mathsf{EXPT}^{\mathsf{ID\text{-}CPA}}_{\mathcal{A},\pi}(\lambda)$ between a challenger and an adversary $\mathcal{A}$ as follows:

$\mathsf{EXPT}^{\mathsf{ID\text{-}CPA}}_{\mathcal{A},\mathsf{IBE}}(\lambda) \to \{0, 1\}$:

Setup: At the beginning of the game, the challenger runs $\mathsf{Setup}(\lambda)$ to obtain master key pair $(\mathsf{mpk}, \mathsf{msk})$, and then it sends public master key mpk to the adversary $\mathcal{A}$.

Phase 1: The adversary may adaptively make secret key queries for the identities in ID. Receiving an identity $\mathsf{id} \in \mathsf{ID}$ as a key query from the adversary, the challenger runs $\mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, \mathsf{id})$ to get a secret key $\mathsf{sk}_{\mathsf{id}}$ for the identity id, and sends it to the adversary.

Chall: After finishing Phase 1, the adversary sends out two equal-length messages μ_0, μ_1 and an identity $\mathsf{id}^* \in \mathsf{ID}$, which is never made a secret key query in previous Phase 1, to the challenger as its challenge. The challenger chooses a random bit $b \in \{0,1\}$, and obtains challenge ciphertext ct^* by encrypting μ_b under the challenge identity id^*, and then sends ct^* to the adversary.

Phase 2: After the challenge query, the adversary $\mathcal{A}$ continues to make an adaptive secret key query for any identity $\mathsf{id} \neq \mathsf{id}^*$. The challenger responds as in Phase 1.

Guess: After all, the adversary outputs an guess b' of b. The experiment outputs 1 if $b = b'$, and 0 otherwise.

Definition 2.2 *We say an* IBE *scheme achieves adaptive security under a chosen message security attack (simply, adaptive security) if for any probabilistic polynomial time (PPT) adversary* $\mathcal{A}$ *and the experiment* $\mathsf{EXPT}^{\mathsf{ID\text{-}CPA}}_{\mathcal{A},\mathsf{IBE}}(\lambda)$ *defined above, the following holds:*

$$\left| \Pr\left[\mathsf{EXPT}^{\mathsf{ID\text{-}CPA}}_{\mathcal{A},\mathsf{IBE}}(\lambda) \to 1\right] - \frac{1}{2} \right| \leq \mathsf{negl}(\lambda),$$

where $\mathsf{negl}(\lambda)$ *is the negligible function in security parameter* λ.

2.2 Truncation Collision-Resistant Hash Function

TCRH. In this paper, We use the truncation collision-resistant hash function from [12] in our construction of a compact IBE scheme. Here, we define the truncation collision-resistant hash function. For an integer n, let $H : \{0,1\}^* \to \{0,1\}^n$ be a hash function, then, for an index $i \in [n]$, we identify the i-th truncation of the hash functions H, denoted H_i, as a function which outputs the prefix of the output of the hash function H of length i. In particular, the n-th truncated hash function H_n is identical to H. Note that a hash function family can be constructed from a hash function by introducing an evaluation key. Namely, we can construct a hash function family $\mathcal{H}$ from a hash function H as follows:

$$\mathcal{H} := \left\{ H_{\mathsf{ek}}, \mathsf{ek} \leftarrow \{0,1\}^\lambda | H_{\mathsf{ek}}(x) := H(x|\mathsf{ek}) \text{ for any } x \in \{0,1\}^* \right\}$$

We formally define a hash function's Truncation Collision Resistance (TCR) security as follows.

Definition 2.3 *We say a hash function family $\mathcal{H}$ is i-th truncation collision resistant if for any adversary $\mathcal{A}$ modeled as a Turing machine, the following probability holds:*

$$\Pr_{H \xleftarrow{} \mathcal{H}} \left[\begin{array}{c} (x_1, x_2, \cdots, x_Q) \leftarrow \mathcal{A}(H) \\ \exists j, k \in [Q], s.t. H_i(x_j) = H_i(x_k) \wedge x_j \neq x_k \end{array} \right] \leq \frac{t_{\mathcal{A}}^2}{2^{i+1}},$$

where H_i is the i-th truncation of H, Q is the number of elements that the algorithm $\mathcal{A}$ able to output, and $t_{\mathcal{A}}$ is the running time of the algorithm $\mathcal{A}$.

Obviously, for a small constant index $i \in [n]$, the i-th truncation collision resistance of any hash function can be trivially broken. This is the case since the algorithm $\mathcal{A}$ can output more than 2^i elements for any constant integer i, and a collision exists in the first i bits. Hence, to make use of the truncation collision property of a hash function, we need to set the index to a super-constant number. A more detailed discussion about the rationality of the above definition can be found in [11].

Partition function, also known as a balanced admissible hash function [10], is vital to the security proof of partition-based IBE schemes. The following definition generalizes the partition function in [26]. Instead of outputting a binary value, the invertibility of the partition function's output in R_q suffices for the security proof of ring lattice-based IBE schemes. Thus, we rewrite the partition function definition from [26] as follows.

Definition 2.4 (Partition Function[[26]]) *Let $\mathcal{F} = \{F_\lambda : \mathcal{K}_\lambda \times \mathcal{X}_\lambda \to R_2\}$ be an ensemble of function family, we say that $\mathcal{F}$ is a partition function, if there exists an efficient algorithm $\mathsf{PrtSmp}(1^\lambda)$, which takes as input polynomial bounded $Q = Q(\lambda) \in \mathbb{N}$ and noticeable $\epsilon = \epsilon(\lambda) \in (0, 1/2]$ and outputs K such that :*

- *There exists $\lambda_0 \in N$ such that $\mathsf{Pr}\left[K \in \mathcal{K}_\lambda : K \xleftarrow{\$} \mathsf{PrtSmp}(1^\lambda, Q(\lambda), \epsilon(\lambda))\right] = 1$ for all $\lambda > \lambda_0$, where λ_0 may depend on $Q(\lambda)$ and $\epsilon(\lambda)$.*
- *For $\lambda > \lambda_0$, there exists $\gamma_{\max}(\lambda)$ and $\gamma_{\min}(\lambda)$ that depends on $Q(\lambda)$ and $\epsilon(\lambda)$ such that for all $X^{(1)}, \cdots, X^{(Q)}, X^* \in \mathcal{X}_\lambda$ with $X^* \notin \left\{X^{(1)}, \cdots, X^{(Q)}\right\}$*

$$\gamma_{\max}(\lambda) \geq \mathsf{Pr}\Big[F_K\left(X^{(1)}\right) \in R_q^*, \cdots, F_K\left(X^{(Q)}\right) \in R_q^* \wedge F_K\left(X^*\right) = 0\Big] \geq \gamma_{\min}(\lambda)$$

holds and the function $\tau(\lambda)$ defined as $\tau(\lambda) := \gamma_{\min} \cdot \epsilon(\lambda) - \frac{\gamma_{\max}(\lambda) - \gamma_{\min}(\lambda)}{2}$ is noticeable. The probability is taken over the random choice of the function key $K \xleftarrow{\$} \mathsf{PrtSmp}(1^\lambda, Q(\lambda), \epsilon(\lambda))$.

We call K the partition key and $\tau(\lambda)$ the quality of the partition function.

The proof of the following lemma can be found in the paper of [3, 16]

Lemma 2.5 (Lemma 8 in [16]) *Let us consider an IBE scheme and an adversary A that breaks the adaptively anonymous security (resp. pseudo-randomness)*

with advantage λ. Let the identity space (resp. input space) be X and consider a map γ that maps a sequence of elements in X to a value in [0,1]. We consider the following experiment. We first execute the security game for $\mathcal{A}$. Let X^* be the challenge identity (resp. challenge input) and $X_1, \cdots, X_Q$ be the identities (resp. inputs) for which key extraction queries (resp. evaluation queries) were made. We denote $\mathbb{X} = (X^*, X_1, \cdots, X_Q)$. At the end of the game, we set $coin' \in \{0,1\}$ as $\hat{b} = b'$ with probability $\gamma(\mathbb{X})$ and $\hat{b} \leftarrow \{0,1\}$ with probability $1 - \gamma(\mathbb{X})$. Then, the following holds

$$\left|\mathsf{Pr}[\hat{b} == b] - \frac{1}{2}\right| \geq \gamma_{\min} \cdot \epsilon - \frac{\gamma_{\max} - \gamma_{\min}}{2}$$

where $\gamma_{\max}$ and $\gamma_{\min}$ are the maximum and the minimum of $\gamma(\mathbb{X})$ taken over all possible $\mathbb{X}$, respectively.

2.3 Gaussians, Rings, and Ring LWE

Lattices. For a full rank matrix $\mathsf{B} = [\mathsf{b}_1, \cdots, \mathsf{b}_n] \in \mathbb{Z}^{m \times n}$, the lattice generated by B is the set $\Lambda = L(\mathsf{B}) := \{\mathsf{B} \cdot \mathsf{c} = \sum_{i=1}^{n} c_i \cdot \mathsf{b}_i | \mathsf{c} = (c_1, \cdots, c_n) \in \mathbb{Z}^n\}$. This paper focuses on a particular family of so-called q-ary integer lattices for some integer q. For the positive integers n, m, q, a matrix $\mathsf{A} \in \mathbb{Z}_q^{n \times m}$ defines two types of following full-rank m-dimensional q-ary lattices as follows:

$$\begin{aligned}\Lambda^{\perp}(\mathsf{A}) :=& \{\mathsf{e} | \mathsf{A} \cdot \mathsf{e} = 0 \mod q\},\\ \Lambda(\mathsf{A}) :=& \{\mathsf{e} | \exists \mathsf{z} \in \mathbb{Z}_q^n \ s.t., \mathsf{A}^{\top} \cdot \mathsf{z} = \mathsf{e} \mod q\}\end{aligned}$$

For any vector $\mathsf{u} \in \mathbb{Z}_q^m$, the "shifted" lattice $\Lambda_{\mathsf{u}}^{\perp}(\mathsf{A})$ is defined as $\Lambda_{\mathsf{u}}^{\perp}(\mathsf{A}) := \{\mathsf{e} | \mathsf{A}^{\top} \cdot \mathsf{e} = \mathsf{u} \mod q\}$.

Gaussians. For a real $s > 0$, the n-dimensional gaussian function with parameter s is defined as $\rho_s(x) := \exp(-\pi \frac{\|\mathsf{x}\|^2}{s^2})$. For any n-dimensional vector c, define the shifted gaussian function by $\rho_{s,\mathsf{c}} := \exp(-\pi \frac{\|\mathsf{x}-\mathsf{c}\|^2}{s^2})$. The discrete gaussian distribution over a lattice coset $\Lambda + \mathsf{u}$ is defined as $D_{\Lambda+\mathsf{u},s}(\mathsf{x}) = \frac{\rho_s(\mathsf{x})}{\rho_s(\Lambda+\mathsf{u})}$. For any positive real $\epsilon_s > 0$, the smoothing parameter of a lattice Λ, denoted $\eta_{\epsilon_s}(\Lambda)$, is the smallest real $s > 0$ such that $\rho_{1/s}(\Lambda^* \setminus \{0\}) \leq \epsilon$. For the Gaussian distribution over a lattice, we have the following tail bound.

Lemma 2.6 ([8,19]) *Let $\Lambda \subset \mathbb{R}^n$ be a lattice and $s > \eta_{\epsilon_s}(\Lambda)$ for some $\epsilon_s \in (0, 1/2)$. For any $\mathsf{c} \in \mathrm{span}(\Lambda)$, we have*

$$\mathsf{Pr}\left[\|D_{\Lambda+\mathsf{c},s}\| \geq s\sqrt{n}\right] \leq 2^{-n} \cdot \frac{1+\epsilon_s}{1-\epsilon_s}.$$

Furthermore, if $\mathsf{c} = 0$, the bound holds for any $r > 0$ with $\epsilon_s = 0$.

Lemma 2.7 (Noise Rerandomization [16]) *Let q, ℓ, m be positive integers and r a positive real satisfying $r > \max\{\eta_{\epsilon_s}(\mathbb{Z}^m), \eta_{\epsilon_s}(\mathbb{Z}^{\ell})\}$. Let $\mathsf{b} \in \mathbb{Z}_q^m$ be arbitrary and x chosen from $D_{\mathbb{Z}^m,r}$. Then for any $\mathsf{V} \in \mathbb{Z}^{m \times \ell}$ and positive real*

$\sigma > \mathsf{s}_1(\mathsf{V})$, *there exists a* PPT *algorithm* $\mathsf{ReRand}(\mathsf{V}, \mathsf{b}+\mathsf{x}, r, \sigma)$ *that outputs* $\mathsf{b}'^{\top} = \mathsf{b}^{\top}\mathsf{V} + \mathsf{x}'^{\top} \in \mathbb{Z}_q^{\ell}$ *where the statistical distance of the discrete Gaussian* $D_{\mathbb{Z}^{\ell}, 2r\sigma}$ *and the distribution of* x' *is within* $8\epsilon_s$.

Rings. We use R to denote the quotient ring $\mathbb{Z}[x]/\Phi_m(x)$ for m-cyclotomic polynomial $\Phi(x)$, and use R_q to denote R/qR for some modulus q. In this paper, we set $m = 2n$ for some n a power of 2, then the cyclotomic polynomial $\Phi_m(x) = x^n + 1$. The following lemmas show the invertibility of the ring elements with small degree and trinary or relatively small coefficients.

Lemma 2.8 ([16]) *Let q be a prime such that $q \equiv 3 \bmod 8$ and n be a power of 2. Let $R_q = \mathbb{Z}_q[x]/\Phi_{2n}(x)$. Then, all non-zero $u \in R_q$ satisfying $\|\mathsf{Coeffs}(u)\|_2 < \sqrt{q}$ are invertible, i.e., $u \in R_q^*$.*

We identify a ring element as a vector in $\mathbb{Z}^n$ by the natural coefficient embedding $\mathsf{Coeffs} : R \to \mathbb{Z}^n$ that maps a ring element $a = \sum_{i=0}^{n-1} a_i \cdot x^i$ to the integer vector $\mathsf{Coeffs}(a) := (a_0, \cdots, a_{n-1}) = \mathsf{a}$. Note that benefited from the special setting above, the ring homomorphism $\mathsf{rot} : R \to \mathbb{Z}^{n\times n}$ maps an element in R to an anti-cyclic matrix in $\mathbb{Z}^{n\times n}$. Namely, for a ring element $a \in R$, $\mathsf{rot}(a) := [\mathsf{Coeffs}(a)|\mathsf{Coeffs}(a \cdot x^1 \mod \Phi(x))|\cdots|\mathsf{Coeffs}(a \cdot x^{n-1} \mod \Phi(x))]^{\top}$. Furthermore, we define the map rot over ring vectors or matrices naturally by applying it entry-wise. In this paper, we use the notion $\mathsf{a} \leftarrow D_{R^k,s}^{\mathsf{Coeffs}}$ to denote the coefficients of the ring vector $\mathsf{a} \in R^k$ is sampled from the Gaussian distribution $D_{\mathbb{Z},s}$. For the sake of simplicity, we may omit the subscript and simply use $\mathsf{a} \leftarrow D_{R^k,s}$.

We identify the norms of a ring element or ring vectors by the norm of corresponding coefficient embeddings. For a matrix R, we define the largest singular value, denoted $s_1(\mathsf{R})$, by $s_1(\mathsf{R}) = \max_{\|\mathsf{x}\|=1} \|\mathsf{R}\cdot\mathsf{x}\|$.

For a set $S \in \mathbb{Z}$, we use S_R (or R_S) to denote the set of ring elements in which the coefficients are in the set S. The notion $x \leftarrow R_S$ means that x is uniformly sampled from the set R_S. In particular, for an integer $\rho > 0$, we can bound the singular value of a random matrix chosen from $[-\rho, \rho]_R^{s\times t}$ as follows.

Lemma 2.9 ([16] Lemma 2) *For a positive integer ρ and a matrix R uniformly chosen from $[-\rho, \rho]_R^{s\times t}$, there exists a constant C ($C \approx 1/2\pi$) such that for any positive number ω, we have*

$$\mathsf{Pr}\left[s_1(\mathsf{R}) \geq C\rho\sqrt{n}\left(\sqrt{s}+\sqrt{t}+\omega\right)\right] = 2e^{-\pi\omega^2}.$$

The following lemma from [16] plays an important role in the security proof of our IBE scheme. The proof can be found in Appendix B.3 of [16].

Lemma 2.10 (Lemma 4 of [16]) *Let n be a power of two, $q > 4n$ be a prime such that $q \equiv 3 \mod 8$. Let ℓ, k', ρ, k be positive integers that $\ell, k' \geq 1$ $k > 2$ and $\rho < \frac{1}{2}\sqrt{q/n}$. Then for $\mathsf{A} \xleftarrow{\$} R_q^{k'\times k}$ and $\mathsf{X} \xleftarrow{\$} [-\rho, \rho]_R^{k\times\ell}$, we have*

$$\Delta\left((\mathsf{A}, \mathsf{AX}), (\mathsf{A}, U(R_q^{k'\times\ell}))\right) \leq \frac{\ell}{2} \cdot \left(\frac{q^{k'}}{(2\rho+1)^k}\right)^{n/2}.$$

Ring LWE. The ring Learning With Errors (RLWE) problem, introduced in [18], is the ring version of the LWE problem [21]. Let R_q be the ring defined above, χ be a distribution over R, and $s \xleftarrow{\$} R_q$, then the $R\text{-}\mathsf{LWE}_{n,q,\chi}$ distribution is the distribution of the pair $(a, b) \in R_q \times R_q$, where a is uniformly sampled from R_q and $b = a \cdot s + e$ for some $e \leftarrow \chi$. Through this $R\text{-}\mathsf{LWE}_{n,q,\chi}$ distribution, given the bounded number of samples, the decision version of the RLWE problem, denoted $R\text{-}\mathsf{DLWE}_{n,\ell,q,\chi}$, is defined as follows.

Definition 2.11 *Given ℓ independent samples $(a_i, b_i)_{i\in[\ell]} \in (R_q \times R_q)^\ell$, the $R\text{-}\mathsf{DLWE}_{n,\ell,q,\chi}$ problem asks to decide if the samples are from the $R\text{-}\mathsf{LWE}_{n,q,\chi}$ distribution or from the uniform distribution over $(R_q \times R_q)^\ell$.*

Although there are several works [18,20] on the hardness of RLWE problems for special kinds of distributions χ, yet in this paper, we consider the case in which χ is Gaussian distribution. In this paper, we use the $R\text{-}\mathsf{DLWE}$ hardness result from [16], and the re-randomization lemma from [16] follows:

Lemma 2.12 (Theorem 1 of [16]) *Let α be the positive real, m be a power of 2, ℓ be an integer, $\Phi(x) = x^n + 1$ be the m-th cyclotomic polynomial where $m = 2n$, and $R = \mathbb{Z}[x]/(\Phi(x))$. Let $q \equiv 3 \mod 8$ be a prime such that there is another prime $p \equiv 1 \mod m$ satisfying $p \leq q \leq 2p$. Let $\sigma_{R\text{-}\mathsf{DLWE}} := \alpha q \geq n^{3/2}\ell^{1/4}\omega(\log^{9/4}(n))$. Then, there is a* PPT *quantum reduction from $\tilde{O}(n/\alpha)$-approximate SIVP(or SVP) to $R\text{-}\mathsf{DLWE}_{n,\ell,q,\chi}$ with $\chi = D^{\mathsf{Coeffs}}_{\mathbb{Z}^n,\sigma_{R\text{-}\mathsf{DLWE}}}$.*

Trapdoors for Rings. For positive integers b and $k > k' \geq \lceil \log(q) \rfloor$, let $\mathsf{g}_b^\top = [1|b|b^2|...|b^{k'}|0] \in R^k$ be the gadget matrix. As stated in the work of [19], this gadget matrix has a public trapdoor $\mathsf{T}_{\mathbf{g}_b}$ with small norm, i.e., $\|\mathsf{T}_{\mathbf{g}_b}\| \leq \sqrt{b^2+1}$. Next, we present several useful sampling algorithms from the work of [16,19].

Lemma 2.13 ([16]) *Let n be a power of 2, q be prime larger than $4n$ such that $q \equiv 3 \mod 8$, b, ρ be positive integers satisfying $\rho < \frac{1}{2}\sqrt{q/n}$, and $\epsilon_s \in (0,1)$ be a small real regarding the smoothing parameter. Furthermore, define $\log_1(\cdot) := \log_2(\cdot)$. There are efficient algorithms such that:*

- $\mathsf{TrapGen}(n, k, \rho, q) \to (\mathsf{a}, \mathsf{T_a})$ ([19], Lemma 5.3): *A randomized algorithm that, when $k \geq 2\log_\rho(q)$, outputs a ring vector $\mathsf{a} \in R^k$ and a matrix $\mathsf{T_a} \in R^{k\times k}$, where $\mathsf{rot}(\mathsf{a}^\top) \in \mathbb{Z}^{n\times nk}$ is full-rank matrix and $\mathsf{rot}(\mathsf{T_a}) \in \mathbb{Z}^{nk\times nk}$ is a basis for $\Lambda^\perp(\mathsf{rot}(\mathsf{a}^\top))$ such that a is statistically close to uniform and $\|\mathsf{rot}(\mathsf{T_a})\|_{\mathsf{GS}} < O\left(b\rho\sqrt{n\log_\rho(q)}\right)$.*
- $\mathsf{SampleLeft}(\mathsf{a}, \mathsf{b}, \mathsf{T_a}, u, \sigma) \to \mathsf{e}$ ([7]): *A randomized algorithm that, on input the vectors $\mathsf{a}, \mathsf{b} \in R^k$, where $\mathsf{rot}(\mathsf{a}^\top), \mathsf{rot}(\mathsf{b}^\top) \in \mathbb{Z}^{n\times nk}$ are full-rank, an element $u \in R_q$, a matrix $\mathsf{T_a}$ such that $\mathsf{rot}(\mathsf{T_a}) \in \mathbb{Z}^{nk\times nk}$ is a basis for $\Lambda^\perp(\mathsf{rot}(\mathsf{a}^\top))$ and Gaussian parameter $\sigma > \|\mathsf{rot}(\mathsf{T_a})\|_{\mathsf{GS}} \cdot \sqrt{\log(2n(1+1/\epsilon_s))/\pi}$, outputs a vector $\mathsf{e} \in R^{2k}$ sampled from a distribution which is statistically close to $D_{\Lambda^\perp_{\mathsf{Coeffs}(u)}\left(\mathsf{rot}([\mathsf{a}^\top|\mathsf{b}^\top])\right),\sigma}$, i.e., $[\mathsf{a}^\top|\mathsf{b}^\top] \cdot \mathsf{e} = u$, and $\mathsf{Coeffs}(\mathsf{e})$ is distributed according to $D_{\Lambda^\perp_{\mathsf{Coeffs}(u)}\left(\mathsf{rot}([\mathsf{a}^\top|\mathsf{b}^\top])\right),\sigma}$.*

– $\mathsf{SampleRight}(\mathsf{a}, \mathsf{R}, u, y, \mathsf{g}_b, \mathsf{T}_{\mathsf{g}_b}, \sigma) \to \mathsf{e}$ *where* $\mathsf{b} = \mathsf{aR} + y \cdot \mathsf{g}_b$ ([3]): *A randomized algorithm that, on input the ring vectors* $\mathsf{a}, \mathsf{g}_b \in R^k$ *such that* $\mathsf{rot}(\mathsf{a}^\top), \mathsf{rot}(\mathsf{g}_b^\top) \in \mathbb{Z}^{n \times nk}$ *are full-rank, elements* $y \in R^*, u \in R$*, a matrix* $\mathsf{R} \in R^{k \times k}$*, a matrix* $\mathsf{T}_{\mathsf{g}_b} \in R^{k \times k}$ *such that* $\mathsf{rot}(\mathsf{T}_{\mathsf{g}_b})$ *is a basis for the lattice* $\Lambda^\perp(\mathsf{rot}(\mathsf{g}_b))$*, and a Gaussian parameter* $\sigma > s_1(\mathsf{R}) \cdot \|\mathsf{rot}(\mathsf{T}_{\mathsf{g}_b})\|_{\mathsf{GS}} \cdot \sqrt{\log(2n(1+1/\epsilon_s))/\pi}$*, outputs a vector* $\mathsf{e} \in R^{2k}$ *sampled from a distribution which is statistically close to* $D_{\Lambda^\perp_{\mathsf{Coeffs}(u)}\left(\mathsf{rot}([\mathsf{a}^\top|\mathsf{b}^\top]\right),\sigma}$*, i.e.,* $[\mathsf{a}^\top|\mathsf{b}^\top] \cdot \mathsf{e} = u$*, and* $\mathsf{Coeffs}(\mathsf{e})$ *is distributed according to* $D_{\Lambda^\perp_{\mathsf{Coeffs}(u)}\left(\mathsf{rot}([\mathsf{a}^\top|\mathsf{b}^\top]\right),\sigma}$.
– ([19]) *Let* $k \geq \lceil \log_b(q) \rceil$*. There is a publicly known matrix* $\mathsf{T}_{\mathsf{g}_b}$ *such that* $\mathsf{rot}(\mathsf{T}_{\mathsf{g}_b})$ *is a basis for the lattice* $\Lambda^\perp(\mathsf{rot}(\mathsf{g}_b^\top))$ *and* $\|\mathsf{rot}(\mathsf{T}_{\mathsf{g}_b})\|_{\mathsf{GS}} \leq \sqrt{b^2+1}$*. Furthermore, there exists a deterministic polynomial time algorithm* g_b^{-1} *which takes input* $\mathsf{u} \in R_q^k$*, and outputs* $\mathsf{R} = \mathsf{g}_b^{-1}(\mathsf{u}^\top)$ *such that* $\mathsf{R} \in [-b, b]_{R^{k \times k}}$*,* $\mathsf{g}_b^\top \cdot \mathsf{R} = \mathsf{u}^\top$*, and* $s_1(\mathsf{R}) \leq nkb$*. Similarly, there exists a randomized polynomial time algorithm* $\widehat{\mathsf{g}_b}^{-1}$ *which takes input* $\mathsf{u} \in R_q^k$*, and outputs* $\mathsf{R} \leftarrow \widehat{\mathsf{g}_b}^{-1}(\mathsf{u}^\top)$ *such that* $\mathsf{g}_b^\top \cdot \mathsf{R} = \mathsf{u}^\top$*. Each coefficient in any entry of* R *follows a sub-Gaussian centered* 0 *with parameter* $O(1)$*, implying* $s_1(\mathsf{R}) \leq \tilde{O}(b\sqrt{nk})$ *with an overwhelming probability.*

2.4 Homomorphic Computations

We defined the encoding of a ring element α as $\mathsf{Encode}(\alpha) := \mathsf{a}^\top \cdot \mathsf{R} + \alpha \cdot \mathsf{g}_b^\top$, where the ring vector a is uniform over R_q^k and $\mathsf{R} \in R_q^{k \times k}$ is a small norm ring matrix. It is easy to see the homomorphic additivity and multiplicativity of this encoding. For a function $f : R^\kappa \to R$, the above encoding enables us to obtain an encoding of the function value $f(\boldsymbol{\alpha})$ for $\boldsymbol{\alpha} \in R^\kappa$. To evaluate the quality of homomorphic evaluation of partition functions, we use the notion of δ-expanding from the previous works [2, 4, 26].

Definition 2.14 *For the function* $f_K : \mathcal{X}^t \to \mathcal{Y} \in \mathbb{Z}$ *and a function family* $\mathcal{F}$ *defined as* $\mathcal{F} := \{f_K : K \in \mathcal{K}\}$*, we say that the two deterministic algorithms* $(\mathsf{PubEval}, \mathsf{TrapEval})$ *are* δ*-expanding with respect to the function* $f_K(\cdot) \in \mathcal{F}$ *if they have following properties:*

- $\mathsf{PubEval}\,(\mathsf{a} \in R_q, f) \to \mathsf{b}_f \in R_q^k$ **:** *on input an encoding vector* $\mathsf{a} \in R_q^k$ *and a function* f*, it outputs a vector* $\mathsf{b}_f \in \mathbb{Z}_q^k$

$$\mathsf{b}_f \leftarrow \mathsf{PubEval}(\mathsf{a}, f)$$

- $\mathsf{TrapEval}\left(\mathsf{a} \in R_q^k, \{\mathsf{R}_i \in R_q^{k \times k}\}_{i \in [t]}, x \in \mathcal{X}, f\right) \to \mathsf{R}_f \in R_q^{k \times k}$ **:** *on input a encoding vector* $\mathsf{a} \in R_q^k$*, trapdoor information* $\mathsf{R}_i \in R_q^{k \times k}$*, an element* $\mathsf{x} \in \mathcal{X}^t$*, and the function* f*, this trapdoor evaluation algorithm outputs a matrix* $\mathsf{R}_f \in \mathbb{Z}_q^{nk \times k}$ *such that*

$$\mathsf{PubEval}(\{\mathsf{a}^\top \cdot \mathsf{R}_i + \mathsf{x}_i \cdot \mathsf{g}_b^\top\}_{i \in [t]}, f) = \mathsf{a}^\top \cdot \mathsf{R}_f + f(\mathsf{x}) \cdot \mathsf{g}_b^\top,$$

and $\|\mathsf{R}_f\| \leq \delta \cdot \max\{\|\mathsf{R}_i\|\}$.

Furthermore, the equality test function defined below plays an important role in our design of the partition function and its homomorphic evaluations.

Definition 2.15 (Equality Test Function) *Define function* $\mathsf{Equal}_\beta(\cdot)$ *parameterized by* $\beta \in [2n]$ *as follows: on input* $x^\alpha \in R$ *for some* $\alpha \in [n]$*, the function outputs* 1 *if* $\alpha \equiv \beta \mod m$ *and* 0 *otherwise.*

The following theorem from [2] shows the existence and quality of the homomorphic evaluation algorithms for the above equality test function.

Lemma 2.16 ([2]) *There are algorithms* $(\mathsf{PubEval}, \mathsf{TrapEval})$ *for the function family* $\{\mathsf{Equal}_\beta\}_{\beta\in[n]}$ *such that they are* $mn(kb)^2$*- expanding in the plain model and* $m\sqrt{nk}b^2k$*- expanding in the CRS model with respect to the function family* $\{\mathsf{Equal}_\beta(\cdot)\}_{\beta\in[m]}$*.*

3 Partitioning via TCRH Functions

In this section, we show our construction of the partitioning function. The key space of our partition function is $\mathcal{K} := \mathbb{Z}^d \times [d]$ for some integer $d > 0$. In other words, any key $K := (t, i)$ is a tuple consists of a λ-bit string (an element in $\mathbb{Z}_2[X]$) in $\{0,1\}^d$ and an index in $[\lambda]$. We continue to identify the bit strings as elements of the binary coefficient ring. For example, for a integer d, a bit string $t = (t_0, t_1, \cdots, t_{d-1}) \in \{0,1\}^d$ can be identified as a ring element in R_q, e.g., $t(x) := \sum_{j=0}^{d-1} t_j \cdot x^j$. We use $t_{\le i}$ to denote the prefix of t of length $i \in [d]$ and $t_{\le i}(x) := \sum_{j=0}^{i-1} t_j \cdot x^j$

Definition 3.1 *Let* n *be a power of* 2*, and* $q > n$ *be a prime such that* $q \equiv 3 \bmod 8$*. Let* $R_q = \mathbb{Z}_q[x]/\Phi_{2n}(x)$ *and* $\mathsf{Hash} : \{0,1\}^\lambda \to \{0,1\}^d$ *be a hash function for a positive integer* $n \ge d \ge \omega(\log(\lambda))$ *and* $\mathcal{K}_\lambda := \{0,1\}^d \times [d]$*, then for a key* $K = (t^*, i^*) \in \mathcal{K}_\lambda$*, we define the keyed function* $F_K : \{0,1\}^\lambda \to R_q$ *as follows:*

$$F_K(X) := F(K, X) := \mathsf{Hash}(X)_{\le i^*}(x) - t^*_{\le i^*}(x).$$

The function family $\mathcal{F}$ *is defined naturally as* $\mathcal{F} := \{F(K, X) | K \in \mathcal{K}_\lambda\}$*.*

The following theorem shows the partition property of the above function.

Theorem 3.2 *The function defined in Definition 3.1 is a partition function if* Hash *is a truncation collision-resistant hash function.*

Proof. To show the theorem, we first define a PrtSmp and then need to show that the PrtSmp satisfies two properties of the partition function in Definition 2.4. Our definition of PrtSmp as follows:

$\mathsf{PrtSmp}(1^\lambda, t_{\mathcal{A}}, \epsilon)$: let $i^* \in [d]$ be the integer such that

$$\frac{1}{2^{i^*+1}} \leq \frac{\epsilon}{2t_{\mathcal{A}}^2} \leq \frac{1}{2^{i^*}},$$

then it selects a random $t^* \xleftarrow{\$} \{0,1\}^d$, and outputs $K = (t^*, i^*)$.

We now show that PrtSmp satisfies the first property in Definition 2.4. When $d = \omega(\log(\lambda))$, $\epsilon(\lambda)$ is noticeable, and $t_{\mathcal{A}}(\lambda)$ is polynomially bounded, then we have $2^d > 2t_{\mathcal{A}}^2/\epsilon(\lambda)$ for sufficiently large λ, and thus there exists an integer $i^* := \left\lfloor \log\left(\frac{2t_{\mathcal{A}}^2}{\epsilon(\lambda)}\right) \right\rfloor <= d$ such that

$$2^{i^*} = 2^{\left\lfloor \log\left(\frac{2t_{\mathcal{A}}^2}{\epsilon(\lambda)}\right)\right\rfloor} \leq 2^{\log\left(\frac{2t_{\mathcal{A}}^2}{\epsilon(\lambda)}\right)} = \left(\frac{2t_{\mathcal{A}}^2}{\epsilon(\lambda)}\right) \leq 2^{\left\lfloor \log\left(\frac{2t_{\mathcal{A}}^2}{\epsilon(\lambda)}\right)\right\rfloor + 1} = 2^{i^*+1}.$$

Therefore, we have that $K = (t^*, i^*) \in \{0,1\}^d \times [d] = \mathcal{K}$ for sufficiently large parameter λ.

Next, we show that PrtSmp satisfies the second property in Definition 2.4. Let $\gamma = \gamma(X^{(1)}, X^{(2)}, \cdots, X^{(Q)}, X^*) := \Pr_K\left[\bigwedge_{i=1}^Q F_K\left(X^{(i)}\right) \in R_q^* \wedge \mathbb{F}_K\left(X^*\right) = 0\right]$, we first show (1) $\gamma_{\max} = \frac{1}{2^{i^*}}$ is an upper bound of $\gamma(X^{(1)}, X^{(2)}, \cdots, X^{(Q)}, X^*)$, and (2) $\gamma_{\min} = \left(1 - \frac{t_{\mathcal{A}}^2}{2^{i^*+1}}\right) \cdot \frac{1}{2^{i^*}}$ is an lower bound of $\gamma(X^{(1)}, X^{(2)}, \cdots, X^{(Q)}, X^*)$.

To show (1), we have that

$$\begin{aligned}
\gamma(X^{(1)}, X^{(2)}, \cdots, X^{(Q)}, X^*) &\leq \Pr_K\left[F_K\left(X^*\right) = 0\right] \\
&= \Pr_K\left[\mathsf{Hash}(X)_{\leq i^*}(x) - t^*_{\leq i^*}(x) = 0\right] \\
&= \Pr_K\left[\mathsf{Hash}(X)_{\leq i^*} = t^*_{\leq i^*}\right] \\
&= \frac{1}{2^{i^*}},
\end{aligned}$$

where the last equality is from the randomness of $t^* \in \{0,1\}^d$.

To show (2), we have the following:

$$\begin{aligned}
&\gamma(X^{(1)}, X^{(2)}, \cdots, X^{(Q)}, X^*) \\
&\quad = \Pr_K\left[\bigwedge_{i=1}^Q F_K\left(X^{(i)}\right) \in R_q^* | F_K\left(X^*\right) = 0\right] \cdot \Pr_K\left[F_K\left(X^*\right) = 0\right] \\
&\quad = \Pr_K\left[\exists j \in [Q], s.t., F_K\left(X^{(j)}\right) \notin R_q^* | F_K\left(X^*\right) = 0\right] \cdot \frac{1}{2^{i^*}} \\
&\quad = \left(1 - \Pr_K\left[\exists j \in [Q], s.t., F_K\left(X^{(j)}\right) \notin R_q^* | F_K\left(X^*\right) = 0\right]\right) \cdot \frac{1}{2^{i^*}}
\end{aligned}$$

Note that the coefficients of $F_K\left(X^{(j)}\right)$, for any $j \in [Q]$, are in $\{-1, 0, 1\}$, and thus the 2-norm of $F_K\left(X^{(j)}\right)$ is smaller than $\sqrt{n} < \sqrt{q}$. Therefore, from

Lemma 2.8, the polynomial $F_K\left(X^{(j)}\right)$ is invertible if it is non-zero. Namely, if $F_K\left(X^{(j)}\right)$ is not invertible, it is a zero polynomial. Therefore, we have

$$\begin{aligned}
&\gamma(X^{(1)}, X^{(2)}, \cdots, X^{(Q)}, X^*) \\
&\quad = \left(1 - \Pr_K\left[\exists j \in [Q], s.t., F_K\left(X^{(j)}\right) = 0 | F_K\left(X^*\right) = 0\right]\right) \cdot \frac{1}{2^{i^*}} \\
&\quad \geq \left(1 - \Pr_K\left[\exists j \in [Q], s.t., F_K\left(X^{(j)}\right) = F_K\left(X^*\right)\right]\right) \cdot \frac{1}{2^{i^*}} \\
&\quad \geq \left(1 - \Pr_K\left[\exists j \in [Q], s.t., \mathsf{Hash}_{i^*}\left(X^{(j)}\right) = \mathsf{Hash}_{i^*}\left(X^*\right)\right]\right) \cdot \frac{1}{2^{i^*}} \\
&\quad \geq \left(1 - \frac{t_{\mathcal{A}}^2}{2^{i^*+1}}\right) \cdot \frac{1}{2^{i^*}},
\end{aligned}$$

where the last inequality is from the TCR property of Hash.

Next we show that $\tau(\lambda) = \gamma_{\min} \cdot \epsilon - \frac{\gamma_{\max} - \gamma_{\min}}{2}$ is noticeable. From the above results, we have that

$$\begin{aligned}
\tau &= \left(1 - \frac{t_{\mathcal{A}}^2}{2^{i^*+1}}\right) \cdot \frac{1}{2^{i^*}} \cdot \epsilon - \left(\frac{1}{2^{i^*}} - \left(1 - \frac{t_{\mathcal{A}}^2}{2^{i^*+1}}\right) \cdot \frac{1}{2^{i^*}}\right) \cdot \frac{1}{2} \\
&= \left(1 - \frac{t_{\mathcal{A}}^2}{2^{i^*+1}}\right) \cdot \frac{1}{2^{i^*}} \cdot \epsilon - \frac{1}{2^{i^*}} \cdot \frac{t_{\mathcal{A}}^2}{2^{i^*+1}} \cdot \frac{1}{2} \\
&\geq \frac{1}{2^{i^*}} \cdot \frac{\epsilon}{2} - \frac{1}{2^{i^*}} \cdot \frac{\epsilon}{4} \\
&\geq \frac{\epsilon}{2t_{\mathcal{A}}^2} \cdot \frac{\epsilon}{4} = \frac{\epsilon^2}{8t_{\mathcal{A}}^2},
\end{aligned}$$

where the inequalities are from the setting of i^*, and the fact that $t_{\mathcal{A}}^2/2^{i^*+1} \leq \epsilon/2 \leq 1/2$. Since ϵ is noticeable and $t_{\mathcal{A}}$ is polynomially bounded, thus $\tau(\lambda)$ is noticeable. This completes the proof. □

3.1 Homomorphic Evaluation of Partition Function

As stated in the introduction, we use the homomorphic equality test function from the work of [2] to do the homomorphic evaluation of our partition function. For a bit string $\mathsf{Hash}(\mathsf{id})$, we use $\mathsf{Hash}(\mathsf{id})_{\leq i}(x)$ to denote the ring element of which coefficients are from the first i bits of the string $\mathsf{Hash}(\mathsf{id})$. Then the public evaluation algorithms for our partition functions are described as follows.

Construction 3.3 *Let* $\mathsf{Hash} : \{0,1\}^* \to \{0,1\}^d$ *be a function, and for* $i \in [d]$, $\mathsf{PubEval}_i^{ET} : R_q^k \to R_q^k$, $\mathsf{TrapEval}_i^{ET} : R_q^{k \times k} \times R \to R_q^{k \times k}$ *be the homomorphic evaluation algorithms, then the public evaluation algorithms* $\mathsf{PubEval} : (R_q^k)^2 \times R_q \to R_q^k$, *and* $\mathsf{TrapEval} : (R_q^k)^2 \times R_q \to R_q^k$, *with respect to the function* $F_K(X)$ *for* $K = (t^*, i^*) \in \{0,1\}^d \times [d]$, *are as follows:*

$\mathsf{PubEval}\left(\mathsf{b}_{t^*}, \mathsf{b}_{i^*}, X\right) \to \mathsf{b}_X$:

1 *Computes the hash value* $t := \mathsf{Hash}(X)$.
2 *Computes* $\mathsf{b}' := \sum_{i=1}^{d} \mathsf{PubEval}_i^{ET}(\mathsf{b}_{i^*}) \cdot t_{\leq i}(x)$.
3 *Let* $\mathsf{b}_X := \mathsf{b}' - \mathsf{b}_{t^*}$, *and outputs* b_X.

$\mathsf{TrapEval}\,(\mathsf{R}_{t^*}, \mathsf{R}_{i^*}, X) \to \mathsf{R}_X$:
1 *Computes the hash value* $t := \mathsf{Hash}(X)$.
2 *Computes* $\mathsf{R}' := \sum_{i=1}^{d} \mathsf{TrapEval}_i^{ET}(\mathsf{R}_{i^*}) \cdot t_{\leq i}(x)$
2 *Computes* $\mathsf{R}_X := \mathsf{R}' - \mathsf{R}_{t^*}$, *and outputs* R_X.

The following theorem shows the homomorphic functionality of the above public evaluation algorithms.

Theorem 3.4 *Let* $\left(\mathsf{PubEval}^{ET}, \mathsf{TrapEval}^{ET}\right)$ *be the homomorphic evaluation algorithms for the equality test function* $\mathsf{Eqtest}_i(\cdot)$ *for* $i \in [d]$ *and* $d < n$, $\mathsf{b}_{t^*}, \mathsf{b}_{i^*}$ *are GSW-type encodings of* $t_{\leq i^*}$, *and* x^{i^*}, *that is* $\mathsf{b}_{t^*} = \mathsf{a} \cdot \mathsf{R}_{t^*} + t^*_{\leq i^*} \mathsf{g}_b$, *and* $\mathsf{b}_{i^*} = \mathsf{a} \cdot \mathsf{R}_{i^*} + x^{i^*} \mathsf{g}_b$, *then the outputs of the algorithms* $(\mathsf{PubEval}, \mathsf{TrapEval})$ *presented in the Construction 3.3 satisfies :*

$$\mathsf{b}_X = \mathsf{a} \cdot \mathsf{R}_X + F_{(t^*, i^*)}(X) \cdot \mathsf{g}_b$$

Proof. From the step 2 of $\mathsf{PubEval}$, we know that

$$\begin{aligned}
\mathsf{b}' &= \sum_{i=1}^{d} \mathsf{PubEval}_i^{ET}(\mathsf{b}_{i^*}) \cdot t_{\leq i}(x) \\
&= \sum_{i=1}^{d} \left(\mathsf{a} \cdot \mathsf{TrapEval}_i^{ET}(\mathsf{R}_{i^*}) + (i \stackrel{?}{=} i^*) \cdot \mathsf{g}_b \right) \cdot t_{\leq i}(x) \\
&= \mathsf{a} \underbrace{\left(\sum_{i=1}^{d} \mathsf{TrapEval}_i^{ET}(\mathsf{R}_{i^*}) \cdot t_{\leq i}(x) \right)}_{=\mathsf{R}'} + \left(\sum_{i=1}^{d} (i \stackrel{?}{=} i^*) \cdot t_{\leq i}(x) \right) \cdot \mathsf{g}_b,
\end{aligned}$$

where the second equality is from the homomorphic evaluation property of the homomorphic equality test functions $\mathsf{PubEval}_i^{ET}$ and $\mathsf{TrapEval}_i^{ET}$; the third equality is from step 2 of the algorithm $\mathsf{TrapEval}$. Furthermore, from the last step of $\mathsf{PubEval}$, we have

$$\begin{aligned}
\mathsf{b}_X = \mathsf{b}' - \mathsf{b}_{t^*} &= \mathsf{a} \cdot \underbrace{(\mathsf{R}' - \mathsf{R}_{t^*})}_{\mathsf{R}_X} + \underbrace{\left(\sum_{i=1}^{d} (i \stackrel{?}{=} i^*) \cdot t_{\leq i}(x) - t^*_{\leq i^*}(x) \right)}_{=F((t^*, i^*), X)} \cdot \mathsf{g}_b \\
&= \mathsf{a} \cdot \mathsf{R}_X + F((t^*, i^*), X) \cdot \mathsf{g}_b,
\end{aligned}$$

where the second equaity is from the above discussion; the last equality is from the step 3 of algorithm $\mathsf{TrapEval}$ and the Definition 3.1. This completes the proof of the theorem. □

The truncation collision resistance of a hash function essentially assumes that no collision attack significantly outperforms the generic birthday algorithm. Truncation collision resistance extends this assumption to prefixes of the hash function output. Therefore, in the real-world application, extendable-output-functions (XOFs) are a plausible candidate to instantiate TCRHFs. The following theorem shows the quality of the above homomorphic evaluation algorithms.

Theorem 3.5 *For the partition function $f \in \mathcal{F}$ defined in Definition 3.1, there exists deterministic public evaluation algorithms* $(\mathsf{PubEval}, \mathsf{TrapEval})$ *(presented in Construction 3.3), which are $O\left(mn(kb)^2 \cdot d^2\right)$-expanding in the plain model, and $O\left(mb^2k\sqrt{nk} \cdot d^2\right)$-expanding in the crs model.*

Proof. Assume the public evaluation algorithms $\left(\mathsf{PubEval}_i^{ET}, \mathsf{TrapEval}_i^{ET}\right)_{i\in[d]}$ for the function family $\{\mathsf{Equal}_i\}_{I\in[n]}$ are δ-expanding, then from the second step of $\mathsf{TrapEval}$, we have that

$$\begin{aligned}\|\mathsf{R}'\| &= \|\sum_{i=1}^{d} \mathsf{TrapEval}_i^{ET}(\mathsf{R}_{i^*}) \cdot t_{\leq i}(x)\| \leq \sum_{i=1}^{d} d \cdot \|\mathsf{TrapEval}_i^{ET}(\mathsf{R}_{i^*})\| \\ &\leq d^2 \cdot \delta \cdot \|\mathsf{R}_{i^*}\|\end{aligned}$$

where the first inequality is from the fact that the coefficients of $t_{\leq i}(x)$ is in $\{0, 1\}$; the second inequality is from the above assumption. Therefore, from the last step of $\mathsf{TrapEval}$, we have that

$$\|\mathsf{R}_X\| \leq \|\mathsf{R}'\| + \|\mathsf{R}_{t^*}\| \leq d^2 \cdot \delta \cdot \max\{\|\mathsf{R}_{i^*}\|, \|\mathsf{R}_{t^*}\}.$$

Therefore, from the Lemma 2.16, we have that

$$\|\mathsf{R}_X\| \leq O(mn \cdot (kb)^2 \cdot d^2) \cdot \max\{\|\mathsf{R}_{i^*}\|, \|\mathsf{R}_{t^*}\}$$

in the plain model, and

$$\|\mathsf{R}_X\| \leq O(m\sqrt{nk} \cdot kb^2 \cdot d^2) \cdot \max\{\|\mathsf{R}_{i^*}\|, \|\mathsf{R}_{t^*}\}$$

in the *CRS* model. This completes the proof of the theorem. □

4 Our IBE Scheme

In this section, we first present our construction of a ring-based IBE scheme. Thereafter, we show the correctness and the security of our scheme. In this section, we work on a prime ring modulus q such that $q \equiv 3 \bmod 8$.

4.1 The Construction

Here we present our construction of an IBE scheme. Let ID be the identity space of the scheme, and Hash be a hash function, the deterministic algorithms $(\mathsf{PubEval}_{\mathsf{IBE}}, \mathsf{TrapEval}_{\mathsf{IBE}})$ are the public evaluation algorithm for the partition function, then the IBE scheme is described as follows.

Construction 4.1

$\mathsf{Setup}(1^\lambda) \to (\mathsf{msk}, \mathsf{mpk})$: *On input the security parameter* λ*, it first sets the parameters* $n, k, q, \rho, \sigma_0, \sigma_1, \sigma$ *as in next Section 4.2, and then runs as follows:*

- *It runs* $\mathsf{TrapGen}(n, k, \rho, q)$ *to get* $\left(\mathsf{a} \in R_q^k, \mathsf{T_a} \in R_q^{k\times k}\right)$.
- *Chooses random vectors* $\mathsf{b}_0, \mathsf{b}_1$ *from* R_q^k*, and a random ring element* $u \xleftarrow{\$} R_q$.
- *It outputs the master secrete key* $\mathsf{msk} = (\mathsf{T_a})$ *and the master public key* $\mathsf{mpk} = (\mathsf{a}, \mathsf{b}_0, \mathsf{b}_1, u)$.

$\mathsf{KeyGen}(\mathsf{msk}, \mathsf{mpk}, \mathsf{id}) \to (\mathsf{sk_{id}})$: *On input the master key pair* $(\mathsf{msk}, \mathsf{mpk})$ *and an identity* $\mathsf{id} \in \mathsf{ID}$*, it runs as follows:*

- *Computes* $\mathsf{b_{id}} = \mathsf{PubEval}\left(\mathsf{b}_0, \mathsf{b}_1, \mathsf{id}\right)$.
- *Runs* $\mathsf{SampleLeft}\left(\mathsf{a}, \mathsf{b_{id}}, \mathsf{T_a}, u, \sigma\right)$ *to get a ring vector* $\mathsf{e} \in R_q^{2k}$ *such that* $[\mathsf{a}^\top | \mathsf{b}_{\mathsf{id}}^\top] \cdot \mathsf{e} = u$.
- *Outputs the* $sk_{\mathsf{id}} = \mathsf{e}$ *as a secret key for the identity* id.

$\mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, \mu) \to \mathsf{ct} = (c_0, c_1)$: *On input the master public key* mpk*, an identity* $\mathsf{id} \in \mathsf{ID}$ *and a message* $\mu \in R_2$*, it runs as follows:*

- *Computes* $\mathsf{b_{id}} = \mathsf{PubEval}\left(\mathsf{b}_0, \mathsf{b}_1, \mathsf{id}\right)$.
- *Samples ring elements* $s \xleftarrow{\$} R_q, e_0 \leftarrow D^{\mathsf{Coeffs}}_{R_q, \sigma_0}$*, and ring vectors* $\mathsf{e}_1, \mathsf{e}_2 \leftarrow D^{\mathsf{Coeffs}}_{R_q^{2k}, \sigma_1}$.
- *Computes* $c_0 = u \cdot s + e_0 + \left\lceil \frac{q}{2} \right\rceil \cdot \mu$*, and* $\mathsf{c}_1 = \begin{bmatrix} \mathsf{a} \\ \mathsf{b_{id}} \end{bmatrix} \cdot s + \begin{bmatrix} \mathsf{e}_1 \\ \mathsf{e}_2 \end{bmatrix}$.
- *Outputs the ciphertext* $\mathsf{ct} = (c_0, \mathsf{c}_1)$.

$\mathsf{Dec}(\mathsf{sk_{id}}, \mathsf{ct}) \to \mu$: *On input the secret key* $\mathsf{sk_{id}} = \mathsf{e}$ *and the ciphertext* $\mathsf{ct} = (c_0, \mathsf{c}_1) \in R_q \times R_q^{2q}$*, it runs as follows:*

- *Computes* $\mu' = \left\lceil \frac{2}{q} \mathsf{Coeffs}\left(c_0 - \mathsf{e}^\top \cdot \mathsf{c}_1\right) \right\rfloor \mod 2$*, where the rounding function* $\lceil * \rfloor$ *applies coefficient-wise.*
- *Outputs the* μ' *as the decrypted message.*

4.2 Parameter Settings

To ensure our IBE in Construction 4.1 reaches the correctness and the adaptive security, the parameters should meet the corresponding requirements as follows:

- For the requirements of $\mathsf{TrapGen}$ algorithm and gadget vector g_b, the parameters should holds $\rho < \frac{1}{2}\sqrt{q/n}$, $k \geq 2\lceil \log_\rho(q) \rceil$, $k \geq \lceil \log_b(q) \rceil$.
- To ensure the existence of such i^* in the definition of PrtSmp in Theorem 3.1, we need to set d such that $2^d > \frac{2t_{\mathcal{A}}^2}{\epsilon}$, where $t_{\mathcal{A}}$ is the running time of the adversary, and λ is the advantage of the adversary.
- To ensure the hardness of R-DLWE problem, we need σ_0 satisfies the condition of the reduction in Lemma 2.12, that is $\sigma_0 \geq n^{3/2}(k+1)^{1/4}\omega(\log^{9/4}(n))$.

– To meet the requirement of ReRand algorithm, the parameters should satisfy: $\sigma_0 \geq \omega\left(\sqrt{\log(n)}\right)$ and $\sigma_1 \geq 2\sigma_0 \cdot \sqrt{s_1^2(\mathsf{R}_{\mathsf{id}^*}) + 1}$.
– To meet the requirement of $\mathsf{SampleLeft}$ algorithm in Lemma 2.13, we need $\sigma > \|\mathsf{rot}(\mathsf{T}_\mathsf{a})\|_{\mathsf{GS}} \cdot \omega\left(\sqrt{\log(n)}\right)$, where $\|\mathsf{rot}(\mathsf{T}_\mathsf{a})\|_{\mathsf{GS}} < O\left(b\rho\sqrt{n\log_\rho(q)}\right)$.
– To meet the requirement of $\mathsf{SampleRight}$ algorithm in Lemma 2.13, we need $\sigma > s_1(\mathsf{R}_{\mathsf{id}}) \cdot \|\mathsf{rot}(\mathsf{T}_{\mathsf{g}_b})\|_{\mathsf{GS}} \cdot \omega\left(\sqrt{\log(n)}\right)$, where $s_1(\mathsf{R}_t)$ satisfies following in the plain model $s_1(\mathsf{R}_t) < \tilde{O}\left(d^2 \cdot (nkb)^2\right) \cdot \|\mathsf{R}\|$, or satisfies the following in the CRS model $s_1(\mathsf{R}_t) < \tilde{O}\left(d^2 n\sqrt{nk}b^2 \cdot k\right) \cdot \|\mathsf{R}\|$.
  Additionally, $\|\mathsf{R}\| \leq C\rho\sqrt{n}\left(2\sqrt{k} + \omega\right)$ with probability $2e^{-\pi\omega^2}$, and we have $\|\mathsf{rot}(\mathsf{T}_{\mathsf{g}_b})\|_{\mathsf{GS}} \leq \sqrt{b^2+1}$.
– For the correctness of our IBE scheme, we need the modulus q satisfy $q \geq O\left(\omega(\log(n)) \cdot \sigma_0 + \sigma\sigma_1 \cdot \omega(\log(n)) \cdot \sqrt{2nk}\right)$ To achieve overwhelming correctness.

Satisfying the above constraints, We set $b = 2$, $\rho = 1$ and $n = \Theta(\lambda)$ $m = O(n\log(n))$, and set other scheme parameters as follows:

$$\begin{aligned} k &= O(\log(n)) & d &= \omega(\log(n)) & \sigma_0 &= \tilde{O}(n^{1.5}) \\ \sigma_1 &= \tilde{O}(n^4) & \sigma &= \tilde{O}(n^{2.5}) & q &= \tilde{O}(n^7) \end{aligned}$$

Below, we present the correctness and security of our IBE scheme under the above parameter setting.

4.3 Analysis of the IBE Scheme

Correctness. The correctness of the above IBE scheme is similar to the correctness of the Dual Regev encryption scheme, yet the security is more complex. The following theorems show the correctness and the adaptive security of our IBE scheme.

Theorem 4.2 *For the ring modulus q, and Gaussian parameters $\sigma, \sigma_0, \sigma_1$ such that $q \geq O\left(\omega(\log(n)) \cdot \sigma_0 + \sigma\sigma_1 \cdot \omega(\log(n)) \cdot \sqrt{2nk}\right)$, the IBE scheme presented in Construction 4.1 is correct.*

Proof. From the description of the encryption algorithm, we know that

$$\begin{aligned} c_0 - \mathsf{e} \cdot \mathsf{c}_1 =& u \cdot s + e_0 + \left\lceil \frac{q}{2} \right\rceil \cdot \mu - \mathsf{e}^\top \cdot \left(\begin{bmatrix} \mathsf{a} \\ \mathsf{b}_{\mathsf{id}} \end{bmatrix} \cdot s - \begin{bmatrix} \mathsf{e}_1 \\ \mathsf{e}_2 \end{bmatrix} \right) \\ =& \left\lceil \frac{q}{2} \right\rceil \cdot \mu + e_0 - \mathsf{e}^\top \cdot \begin{bmatrix} \mathsf{e}_1 \\ \mathsf{e}_2 \end{bmatrix} \end{aligned}$$

Note that e_0 is sampled from the Gaussian distribution $D^{\mathsf{Coeffs}}_{R_q,\sigma_0}$, thus from the Gaussian tail bound (Lemma 2.6), we have that $\mathsf{Pr}[\|e_0\|_\infty \leq \omega(\log(n)) \cdot \sigma_0] \leq \mathsf{negl}(\lambda)$. Furthermore, we have the following:

$$\begin{aligned}\left\|\mathsf{e}^\top \cdot \begin{bmatrix}\mathsf{e}_1\\ \mathsf{e}_2\end{bmatrix}\right\|_\infty &= \left\|\mathsf{Coeffs}(\mathsf{e}^\top) \cdot \mathsf{rot}\left(\begin{bmatrix}\mathsf{e}_1\\ \mathsf{e}_2\end{bmatrix}\right)\right\|_\infty = \max_{j\in[n]}\left\|\mathsf{Coeffs}(\mathsf{e}^\top) \cdot \mathsf{rot}\left(\begin{bmatrix}\mathsf{e}_1\\ \mathsf{e}_2\end{bmatrix}\right)_j\right\|_\infty \\ &\leq \max_{j\in[n]} \|\mathsf{Coeffs}(\mathsf{e}^\top)\| \cdot \left\|\mathsf{rot}\left(\begin{bmatrix}\mathsf{e}_1\\ \mathsf{e}_2\end{bmatrix}\right)_j\right\|_\infty \\ &\leq \sqrt{2nk}\sigma \cdot \omega(\log(n)) \cdot \sigma_1,\end{aligned}$$

where the last inequality holds except with negligible probability by Lemma 2.6. Thus from the above results, we have that

$$\mathsf{Pr}\left[\left\|e_0 - \mathsf{e}^\top \cdot \begin{bmatrix}\mathsf{e}_1\\ \mathsf{e}_2\end{bmatrix}\right\|_\infty \leq \sqrt{n}\sigma_0 + \sigma\sigma_1 \cdot \omega(\log(n)) \cdot \sqrt{2nk}\right] \leq \mathsf{negl}(\lambda).$$

Thus, from the description of the decryption algorithm, it decrypts correctly except with negligible probability, and thus, the IBE scheme is correct.

□

Theorem 4.3 *The* IBE *scheme described in Construction 4.1 achieves adaptive security if the hash function* Hash *has TCR property and the underlying problem* R-$\mathsf{DLWE}_{n,k+1,q,\sigma_0}$ *is hard.*

Proof. To show the theorem, we prove that if there is an adversary $\mathcal{A}$ who breaks the IBE scheme in the Construction 4.1 with noticeable probability ϵ in a polynomial running time of $t_{\mathcal{A}}$, then there is an algorithm $\mathsf{Sim}_{R\text{-}\mathsf{DLWE}}$ which invokes $\mathcal{A}$ to solve the R-$\mathsf{DLWE}_{n,k+1,q,\sigma_0}$ problem with noticeable probability in polynomial time as well. This contradicts the theorem assumption, and thus, the theorem follows. Before presenting the concrete description of $\mathsf{Sim}_{R\text{-}\mathsf{DLWE}}$, we first define and analyse hybrid games, from which the design idea of $\mathsf{Sim}_{R\text{-}\mathsf{DLWE}}$ can be easily seen. The hybrid games are defined as follows.

Game_0 : This is the original adaptive IBE security game between the adversary $\mathcal{A}$ and the challenger Chall.

Game_1 : In this game, we change the Game_0 so that the challenger Chall procedure has a conditional abort at the end of the game. Namely, this game is identical to the Game_0 except that, at the end of the game, the challenger runs $\mathsf{PrtSmp}(1^\lambda, t_{\mathcal{A}}, \epsilon) \to K = (t^*, i^*)$ and *abort* the game if the following condition holds;

$$\bigwedge_{i=1}^{Q} \left(F\left(K, id^i\right) \in R_q^*\right) \bigwedge F\left(K, id^*\right) = 0,$$

where Q is the upper bound of the number of identities that the adversary asked for key extraction queries, and $\mathsf{id}^1, \cdots, \mathsf{id}^Q$ are the identities for which $\mathcal{A}$ has made key extraction queries, and id^* is the challenge identity.

Game_2 : In this game, we change the way that the ring vectors $\mathsf{b}_0, \mathsf{b}_1$ are generated. Here, instead of sampling $\mathsf{b}_0, \mathsf{b}_1$ uniformly over R_q^k, the challenger first samples $\mathsf{R}_0, \mathsf{R}_1 \leftarrow R_{[-\rho,\rho]}^{k\times k}$, and sets

$$\mathsf{b}_0 := \mathsf{a}^\top \cdot \mathsf{R}_0 + t^*_{\leq i^*}(x) \cdot \mathsf{g}_b,\ \mathsf{b}_1 := \mathsf{a}^\top \cdot \mathsf{R}_1 + x^{i^*} \cdot \mathsf{g}_b,$$

Game_3 : This game is the same as Game_2 except that we change way of generating public ring vector a and the key extraction procedure. Here, the challenger samples $\mathsf{a} \xleftarrow{\$} R_q^k$ uniformly instead of generating it with a trapdoor by running the trapdoor generating algorithm $\mathsf{TrapGen}$. To extract a secret key for an identity id, the challenger first computes

$$\mathsf{R}_{\mathsf{id}} = \mathsf{TrapEval}\left(\mathsf{R}_0, \mathsf{R}_1, \mathsf{id}\right).$$

Then sample a ring vector $\mathsf{e}_{\mathsf{id}} \in R_q^{2k}$ by running $\mathsf{SampleRight}$ as

$$\mathsf{e}_{\mathsf{id}} = \mathsf{SampleRight}\left(\mathsf{a}, \mathsf{R}_{\mathsf{id}}, u, F_{\mathsf{id}}(t^*, i^*), \mathsf{g}_b, \mathsf{T}_{\mathsf{g}_b}, \sigma\right).$$

Finally, the challenger outputs e_{id} as a corresponding key for the identity id.

Game_4 : In this game, we change the way that the challenge ciphertext $\mathsf{ct}^* = (c_0^*, \mathsf{c}_1^*)$ is generated. Here, the challenger first selects $s \xleftarrow{\$} R_q$ and $\mathsf{x} \leftarrow D^{\mathsf{Coeffs}}_{R^k,\sigma_0}$, and compute $\mathsf{v} = \mathsf{a}\cdot s + \mathsf{x}$. Then the challenger samples $e_0 \leftarrow D^{\mathsf{Coeffs}}_{R,\sigma_0}$, and generates the challenge ciphertext as follows:

$$c_0^* = u \cdot s + e_0 + \left\lceil \frac{q}{2} \right\rceil \cdot \mu,\ ,\ \mathsf{c}_1^* = \mathsf{ReRand}\left([I_k|\mathsf{R}_{\mathsf{id}^*}]^\top, \mathsf{v}, \sigma_0, \frac{\sigma_1}{2\sigma_0}\right),$$

where I_k is the identity matrix.

Game_5 : In this game, we change the way that the challenge ciphertext $\mathsf{ct}^* = (c_0^*, \mathsf{c}_1^*)$ is generated. Here, instead of masking the message with LWE samples, the challenger masks the message with uniformly random samples. Formally, the challenger first samples $u_0 \leftarrow R_q$, $\mathsf{v}' \leftarrow R_q^k$ and $\mathsf{x} \leftarrow D^{\mathsf{Coeffs}}_{R^k,\sigma_0}$, then set $\mathsf{v} = \mathsf{v}' + \mathsf{x}$ and compute the challenge ciphertext as follows:

$$c_0^* = u_0 + \left\lceil \frac{q}{2} \right\rceil \cdot \mu,\ ,\ \mathsf{c}_1^* = \mathsf{ReRand}\left([I_k|\mathsf{R}_{\mathsf{id}^*}]^\top, \mathsf{v}, \sigma_0, \frac{\sigma_1}{2\sigma_0}\right),$$

For $i \in \{0,1,2,3,4,5\}$, let E_1 be the event that the adversary $\mathcal{A}$ wins in Game_i, and let abort be the event that challenger aborts in Game_1. From the assumption, we have that $\mathsf{Pr}[E_0] = \epsilon$. Additionally, from the description of Game_5, it is easy to see that c_0 is independent of c_1. From the uniformity of u_0, the advantage of any adversary in Game_5 is exactly 0, that is $\mathsf{Pr}[E_5] = 1/2$. Furthermore, we have the following results.

Lemma 4.4 $|\mathsf{Pr}[E_1] - 1/2|$ *is noticeable if* ϵ *is noticeable.*

Proof. Since the $\mathcal{A}$'s advantage ϵ is noticeable and the running time $T_{\mathcal{A}}$ is polynomially bounded, the Lemma 2.5 and the second property of the partition function implies that

$$|\mathsf{Pr}[E_1] - 1/2| \geq \gamma_{\min} \cdot \epsilon - \frac{1}{2}(\gamma_{\max} - \gamma_{\min}) \geq \frac{\epsilon^2}{8t_{\mathcal{A}}}$$

, and the probability is noticeable. This completes the proof of Lemma4.4 □

Lemma 4.5 $|\mathsf{Pr}[E_2] - \mathsf{Pr}[E_1]| \leq \mathsf{negl}(\lambda)$

Proof. Recall that the difference between Game_1 and Game_2 is the way of generating the public ring vectors b_0 and b_1. Namely, the two games generate the public ring vectors as follows:

$$\mathsf{Game}_1 : \left(\mathsf{b}_0 \leftarrow R_q^k, \mathsf{b}_1 \leftarrow R_q^k\right)$$
$$\mathsf{Game}_2 : \left(\mathsf{b}_0 = \mathsf{a}^\top \cdot \mathsf{R}_0 + t^*_{\leq i^*}(x) \cdot \mathsf{g}_b, \mathsf{b}_1 = \mathsf{a}^\top \cdot \mathsf{R}_1 + x^{i^*} \cdot \mathsf{g}_b\right)$$

where $R_0, R_1 \leftarrow R^{k \times k}_{-\rho,\rho}$. Thus, Lemma 2.10 shows that the difference between the above two games is negligible. This completes the proof of Lemma 4.5. □

Lemma 4.6 $|\mathsf{Pr}[E_3] - \mathsf{Pr}[E_2]| \leq \mathsf{negl}(\lambda)$

Proof. Note that, in Game_3, the challenger changed the way of generating the public ring vector a and the key extraction procedure. So, to show this lemma, it suffices to show that (1) the difference between the generating procedure of a is negligible, and (2) the difference between the key extraction procedure is negligible. Thus, applying the probability union bound, we complete the proof of this lemma.

Now we show the statement (1): Note that public ring vector a is sampled uniformly from the ring R_q^k in Game_3 instead of by running $\mathsf{TrapGen}$ as in Game_2. From the first property of Lemma 2.13, the difference between the two cases is negligible in security parameter λ.

Next we show the statement (2):Note that in Game_2, the challenger uses the trapdoor information of a to extract the identity key by running $\mathsf{SampleLeft}$ algorithm of Lemma 2.13. However, in Game_3, the challenger has no trapdoor information about the ring vector a. Yet, it can use R_{id} to generate the corresponding secret identity key by running the $\mathsf{SampleRight}$ algorithm of Lemma 2.13. The second and third property of Lemma 2.13 shows that the outputs of the above two sampling algorithms are statistically close to $D_{[\mathsf{a}^\top | \mathsf{b}_{\mathsf{id}}^\top], u, \sigma}$. Thus, the difference between the above two procedures of generating identity keys is negligible. □

Lemma 4.7 $|\mathsf{Pr}[E_4] - \mathsf{Pr}[E_3]| \leq \mathsf{negl}(\lambda)$

Proof. Note that the only difference between the Game_3 and Game_4 is the way the challenger generates the challenge ciphertext $\mathsf{ct} = (c_0, \mathsf{c}_1$. In Game_3, the

challenge ciphertext is generated by the encryption algorithm, while in $game_4$, c_0 is generated by the same way as in Game_3, yet $\mathsf{c}_1 = \mathsf{ReRand}\left([I_k|\mathsf{R}_{\mathsf{id}^*}, \mathsf{v}, \sigma_0, \frac{\sigma}{2\sigma_0}]\right)$. From Lemma 2.7, we have that the difference between the above two games is negligible.

□

Lemma 4.8 $|\mathsf{Pr}[E_5] - \mathsf{Pr}[E_4]| \leq \mathsf{negl}(\lambda)$ *if the* R-$\mathsf{DLWE}_{q,n,\sigma_0,k,k+1}$ *problem is hard.*

Proof. To show this lemma, we construct a simulation algorithm $\mathsf{Sim}_{R\text{-}\mathsf{DLWE}}$ (which is desired in the Theorem 4.3) which tries to solve the R-DLWE problem by invoking an algorithm $\mathcal{B}$ who tries to distinguish Game_4 and Game_5 . The description of $\mathsf{Sim}_{R\text{-}\mathsf{DLWE}}$ is as follows:

$\mathsf{Sim}_{R\text{-}\mathsf{DLWE}}\left((\mathsf{a}^\top|u), (\mathsf{b}^\top, b_0)\right) \rightarrow \{0,1\}$:

On input the $(k+1)$ R-$\mathsf{DLWE}_{n,q,k_1,\sigma_0}$ challenge samples $\left([\mathsf{a}\top|u], [\mathsf{b}^\top, b_0]\right)$, algorithm $\mathsf{Sim}_{R\text{-}\mathsf{DLWE}}$ simulates the Game_4 for $\mathcal{B}$ except the challenge ciphertext respond. It first lets a as the part of the public key and generates the two ring vectors $\mathsf{b}_0, \mathsf{b}_1$ as in Game_4, and let $\mathsf{mpk} = (\mathsf{a}, \mathsf{b}_0, \mathsf{b}_1, u)$ as the master public key of the IBE scheme. The identity key extraction procedure is the same as in Game_4, yet it generates the challenge ciphertext as follows:

$$c_0^* = b_0 + \lceil \frac{q}{2} \rceil \cdot \mu \ , \ \mathsf{c}_1^* = \mathsf{ReRand}\left([I_k|\mathsf{R}_{\mathsf{id}^*}], \mathsf{b}, \sigma_0, \frac{\sigma}{2\sigma_0}\right).$$

If the algorithm $\mathcal{B}$ outputs 1, meaning that the simulated game is Game_4, then the algorithm $\mathsf{Sim}_{R\text{-}\mathsf{DLWE}}$ outputs 1 implies the challenge samples are from $\mathsf{Sim}_{R\text{-}\mathsf{DLWE}}$ distribution. If $\mathcal{B}$ outputs 0, meaning that the simulated game is Game_5, then the algorithm $\mathsf{Sim}_{R\text{-}\mathsf{DLWE}}$ outputs 0, implies the challenge samples are from a uniform distribution.

It is not hard to see that if the given challenge samples $\left([\mathsf{a}\top|u], \mathsf{b}^\top, b_0\right)$ are from R-DLWE distribution, then the algorithm $\mathsf{Sim}_{R\text{-}\mathsf{DLWE}}$ exactly simulates the Game_4; if the given challenge samples are from uniform distribution, then the algorithm $\mathsf{Sim}_{R\text{-}\mathsf{DLWE}}$ exactly simulates the Game_5. Thus the advantage of the simulation algorithm $\mathsf{Sim}_{R\text{-}\mathsf{DLWE}}$ is exactly the advantage of the distinguishing advantage of $\mathcal{B}$. From the hardness assumption of the R-DLWE problem, the advantage of $\mathsf{Sim}_{R\text{-}\mathsf{DLWE}}$ is negligible, and thus, the distinguishing advantage of any algorithm $\mathcal{B}$ is negligible. This completes the proof of Lemma 4.8 □

Put all together: Combining the above results, we have $\mathsf{Pr}[E_5] = 1/2$ and

$$\begin{aligned}
|\mathsf{Pr}[E_5] - 1/2| &= \Big| \sum_{i=1}^{4} (\mathsf{Pr}[E_{i+1}] - \mathsf{Pr}[E_i]) + \mathsf{Pr}[E_1] - 1/2 \Big| \\
&\geq \mathsf{Pr}[E_1] - 1/2| - \sum_{i=1}^{4} |\mathsf{Pr}[E_{i+1}] - \mathsf{Pr}[E_i]| \\
&\geq \frac{\epsilon^2}{8T_{\mathcal{A}}^2} - \mathsf{negl}(\lambda),
\end{aligned}$$

This contradicts the assumption that ϵ is noticeable and $T_{\mathcal{A}}$ is polynomially bounded. This completes the proof of Theorem 4.3. □

Remark 4.9 *We use the Waters' artificial abort technique in the security proof of our IBE scheme, and thus, we come up with a quadratic reduction loss on ϵ. Note that the recent work of [9] applied Bonferroni's inequality to the estimation of the failure probability in the reduction used in Waters' original proof related to the artificial abort stage, and reduced the term ϵ^2 in the reduction loss to $\epsilon^{1.5}$. We remark that the same technique can be used in our proof (in Lemma 4.4) to improve the reduction loss.*

Note that the ratio $1/\alpha = q/\sigma$ directly reflects the hardness of R-DLWE problems. Thus, from the parameter setting in the previous section, we have the following result for the IBE scheme in Construction 4.1.

Corollary 4.10 *If there exists a polynomial time adversary $\mathcal{A}$ who breaks the IBE scheme in Construction 4.1 under our recommended parameters with noticeable probability, then there is a polynomial time algorithm $\mathsf{Sim}_{R\text{-DLWE}}$ who breaks the R-$\mathsf{DLWE}_{n,k+1,q,\sigma_0}$ problem for $\frac{1}{\alpha} = \tilde{O}\left(n^{5.5}\right)$.*

5 Conclusion

In this paper, we presented a compact lattice-based IBE scheme based on the ring LWE assumption and truncation-collision-resistant hash functions (TCRHF). Recall that the master public key of the previous IBE construction of Jager et al. (PKC 2021) from the TCRHFs contains $\omega(\log(n))$ matrices (ring vectors in the ring setting). Our IBE is compact and thus significantly improves the previous work. However, we also note that our IBE construction relies on the TCRHF as the [11]. Therefore, constructing an adaptively secure compact IBE in the standard model from standard lattice problems is still an interesting open problem.

Acknowledgment: We want to thank the anonymous reviewers of SAC 2026 for their insightful comments.

References

1. P. Abla. Identity-based encryption from LWE with more compact master public key. In E. Oswald, editor, *CT-RSA 2024*, volume 14643 of *LNCS*, pages 319–353. Springer, Cham, May 2024. 2, 3
2. P. Abla, F. Liu, H. Wang, and Z. Wang. Ring-based identity based encryption - asymptotically shorter MPK and tighter security. In K. Nissim and B. Waters, editors, *Theory of Cryptography - 19th International Conference, TCC 2021, Raleigh, NC, USA, November 8-11, 2021, Proceedings, Part III*, volume 13044 of *Lecture Notes in Computer Science*, pages 157–187. Springer, 2021. 2, 3, 4, 5, 12, 13, 15

3. S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 553–572. Springer, Berlin, Heidelberg, May / June 2010. 2, 3, 4, 8, 12
4. D. Apon, X. Fan, and F. Liu. Vector encoding over lattices and its applications. *IACR Cryptol. ePrint Arch.*, page 455, 2017. 2, 12
5. M. Bellare and T. Ristenpart. Simulation without the artificial abort: Simplified proof and improved concrete security for Waters' IBE scheme. In A. Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 407–424. Springer, Berlin, Heidelberg, Apr. 2009. 2
6. X. Boyen and Q. Li. Towards tightly secure lattice short signature and id-based encryption. In J. H. Cheon and T. Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 404–434. Springer, Berlin, Heidelberg, Dec. 2016. 2, 3
7. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 523–552. Springer, Berlin, Heidelberg, May / June 2010. 2, 11
8. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In R. E. Ladner and C. Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008. 2, 9
9. G. Hanaoka, S. Katsumata, K. Kimura, K. Takemure, and S. Yamada. Tighter adaptive ibes and vrfs: Revisiting waters' artificial abort. In E. Boyle and M. Mahmoody, editors, *Theory of Cryptography - 22nd International Conference, TCC 2024, Milan, Italy, December 2-6, 2024, Proceedings, Part III*, volume 15366 of *Lecture Notes in Computer Science*, pages 124–155. Springer, 2024. 2, 24
10. T. Jager. Verifiable random functions from weaker assumptions. In Y. Dodis and J. B. Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 121–143. Springer, Berlin, Heidelberg, Mar. 2015. 8
11. T. Jager and R. Kurek. Short digital signatures and ID-KEMs via truncation collision resistance. In T. Peyrin and S. Galbraith, editors, *ASIACRYPT 2018, Part II*, volume 11273 of *LNCS*, pages 221–250. Springer, Cham, Dec. 2018. 2, 3, 8, 24
12. T. Jager, R. Kurek, and D. Niehues. Efficient adaptively-secure IB-KEMs and VRFs via near-collision resistance. In J. Garay, editor, *PKC 2021, Part I*, volume 12710 of *LNCS*, pages 596–626. Springer, Cham, May 2021. 2, 3, 5, 7
13. W. Ji, Z. Wang, L. Lyu, and D. Gu. Adaptively secure IBE from lattices with asymptotically better efficiency. In T. Jager and J. Pan, editors, *Public-Key Cryptography - PKC 2025 - 28th IACR International Conference on Practice and Theory of Public-Key Cryptography, Røros, Norway, May 12-15, 2025, Proceedings, Part I*, volume 15674 of *Lecture Notes in Computer Science*, pages 91–124. Springer, 2025. 2
14. W. Ji, Z. Wang, L. Lyu, and D. Gu. Revisiting adaptively secure IBE from lattices with smaller modulus: A conceptually simple framework with low overhead. *IACR Cryptol. ePrint Arch.*, page 1466, 2025. 2
15. S. Katsumata. On the untapped potential of encoding predicates by arithmetic circuits and their applications. In T. Takagi and T. Peyrin, editors, *ASIACRYPT 2017, Part III*, volume 10626 of *LNCS*, pages 95–125. Springer, Cham, Dec. 2017. 2, 3
16. S. Katsumata and S. Yamada. Partitioning via non-linear polynomial functions: More compact IBEs from ideal lattices and bilinear maps. In J. H. Cheon and T. Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 682–712. Springer, Berlin, Heidelberg, Dec. 2016. 2, 3, 8, 9, 10, 11

17. Q. Lai, F.-H. Liu, and Z. Wang. Almost tight security in lattices with polynomial moduli - PRF, IBE, all-but-many LTF, and more. In A. Kiayias, M. Kohlweiss, P. Wallden, and V. Zikas, editors, *PKC 2020, Part I*, volume 12110 of *LNCS*, pages 652–681. Springer, Cham, May 2020. 2, 3
18. V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 1–23. Springer, Berlin, Heidelberg, May / June 2010. 11
19. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, Berlin, Heidelberg, Apr. 2012. 4, 9, 11, 12
20. C. Peikert, O. Regev, and N. Stephens-Davidowitz. Pseudorandomness of ring-LWE for any ring and modulus. In H. Hatami, P. McKenzie, and V. King, editors, *49th ACM STOC*, pages 461–473. ACM Press, June 2017. 11
21. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In H. N. Gabow and R. Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005. 11
22. P. W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th FOCS*, pages 124–134. IEEE Computer Society Press, Nov. 1994. 2
23. B. Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 619–636. Springer, Berlin, Heidelberg, Aug. 2009. 2
24. B. R. Waters. Efficient identity-based encryption without random oracles. In R. Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127. Springer, Berlin, Heidelberg, May 2005. 2
25. S. Yamada. Adaptively secure identity-based encryption from lattices with asymptotically shorter public parameters. In M. Fischlin and J.-S. Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 32–62. Springer, Berlin, Heidelberg, May 2016. 2
26. S. Yamada. Asymptotically compact adaptively secure lattice IBEs and verifiable random functions via generalized partitioning techniques. In J. Katz and H. Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 161–193. Springer, Cham, Aug. 2017. 2, 3, 4, 8, 12
27. J. Zhang, Y. Chen, and Z. Zhang. Programmable hash functions from lattices: Short signatures and IBEs with small key sizes. In M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 303–332. Springer, Berlin, Heidelberg, Aug. 2016. 2, 3