Ring-based Identity Based Encryption – Asymptotically Shorter MPK and Tighter Security

Parhat Abla[1,2], Feng-Hao Liu[3], Han Wang (Corresponding Author)[1,2], Zhedong Wang[4]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Science, Beijing, China. {parhat,wanghan}@iie.ac.cn.
[2] School of Cyber Security, University of Chinese Academy of Science, Beijing, China.
[3] Florida Atlantic University, Boca Raton, FL, USA. liuf@fau.edu.
[4] School of Cyber Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. Zhedw1991@gmail.com.

Abstract. This work constructs an identity based encryption from the ring learning with errors assumption (RLWE), with shorter master public keys and tighter security analysis. To achieve this, we develop three new methods: (1) a new homomorphic equality test method using nice algebraic structures of the rings, (2) a new family of hash functions with natural homomorphic evaluation algorithms, and (3) a new insight for tighter reduction analyses. These methods can be used to improve other important cryptographic tasks, and thus are of general interests.
Particularly, our homomorphic equality test method can derive a new method for packing/unpacking GSW-style encodings, showing a new non-trivial advantage of RLWE over the plain LWE. Moreover, our new insight for tighter analyses can improve the analyses of all the currently known partition-based IBE designs, achieving the best of the both from prior analytical frameworks of Waters (Eurocrypt '05) and Bellare and Ristenpart (Eurocrypt '09).

1 Introduction

Identity-based Encryption (IBE) was introduced by [33] as a generalization of the traditional public-key encryption (PKE) in which a publicly known string (id) of a party can serve as its public key $\mathsf{pk}_{\mathsf{id}}$. This primitive is particularly useful in scenarios that require to manage a large amount of public keys, without the need to access a public-key infrastructure (PKI). Since its first realization [11], there has been significant research in the past two decades [1, 4, 9, 10, 19, 20, 25, 36–40], constructing various IBE schemes from different assumptions.

There have been two major security notions – selective security and adaptive security studied in the literature, where the former requires the adversary to choose the challenge id before seeing the master public key, yet the latter does not have this restriction. Obviously the adaptive security is more desirable by providing stronger security for more realistic settings, yet realizing such a notion is quite challenging, especially when one aims at comparable efficiency in the plain model with the selectively secure designs.

Prior constructions from bilinear groups have achieved this task via the powerful framework of dual-system [36]. However, it is elusive whether the dual-system framework can be instantiated from other assumptions, especially from a post-quantum candidate such as lattices. For the post-quantum settings, even though there are adaptively secure lattice-based IBE, the current instantiations come at a rather higher cost in the size of mpk, ciphertext, and/or larger security loss in the reduction. How to improve these aspects is an important step towards realizing a practical post-quantum IBE.

In this work, we focus on adaptively secure lattice-based IBE with smaller mpk, comparable ciphertexts, and smaller security loss for the reduction. Below we discuss challenges for current approaches and then our new ideas.

Challenges in Current Techniques. Among the existing lattice-based IBE schemes, the most efficient one is the selectively secure scheme by [1], which only requires 2 public matrices in mpk (or ring vectors using Ring-LWE [27]) and has rather small ciphertexts. To achieve the adaptive security, there have been several proposals, but they all have various drawbacks as stated below.

There are two ways to achieve the shortest mpk that exactly matches the selectively secure one as [1], but both suffer from serious issues. The first one simply applies the generic complexity leveraging argument, yet the security reduction would lose 2^ℓ in advantage (ℓ is the bit length of ID), resulting in a much larger security parameter required in the underlying assumption. The second method is a new bootstrapping via a recent technique by [14, 18], which transforms any selectively secure IBE into an adaptively secure one without blowing up the mpk at all. The resulting scheme is however, not considered even close to practical as each ciphertext consists of ℓ garbled circuits.

More efficient IBE can be achieved via a lattice vanishing technique by [1], yet the scheme has a larger mpk (i.e., $O(\lambda)$ basic matrices or ring vectors) and reduction running time (an additive $O(1/\epsilon^2)$ increase), compared with the selectively secure scheme.[5] Later, subsequent work [4, 25, 38, 39] improved this technique by using homomorphic computation in novel ways [3, 21, 29] with more delicate security analyses. Yet these schemes still have several critical shortcomings.

- The best scheme (asymptotically) is the one by [39], which only has $\omega(\log \lambda)$ basic matrices (or ring vectors) in mpk and rather small ciphertexts. However, the IBE construction requires to use Barrington's Theorem [5] to compute an NC1 boolean circuit, which can be done in polynomial time in theory yet would not be expected to be efficient in practice. In fact, the work [39] did not (was not able to) present an explicit construction, making it hard to determine concrete bounds for the parameters for comparison.
- The follow up works [4, 25, 38, 39] removed the $O(1/\epsilon^2)$ blowup of [1] in the reduction running time, but would incur an additional reduction loss of $O(\epsilon)$, multiplicatively. Seemingly this tradeoff is inherent, i.e., the reduction either blows up its running time by $O(1/\epsilon^2)$ additively or loses its advantage by an extra $O(\epsilon)$ multiplicatively, under the current techniques.

[5] λ is the security parameter and ϵ is the adversary's advantage in attacking the IBE scheme.

1.1 Our Contributions

In this work, we significantly improve existing lattice-based IBE in the parameters and security analysis. The crux relies on new techniques related to homomorphic computation in the cyclotomic rings and new analytical insights to achieve tighter analysis for general partition-based IBE. We believe that these tools can be applied broadly and thus are of general interests. Below we summarize our two major contributions, and present our new techniques in Section 1.2.

- We construct an adaptively secure IBE based on Ring-LWE, with $\omega(1)$ ring vectors in the master public key. This improves the prior state-of-the-art [39] by a factor of $\log \lambda$. Additionally, every component in our construction is explicit, i.e., without relying the Barrington's Theorem as required by [39], and thus we are able to determine concrete bounds for all parameters.
- We identify an analytical insight that improves all (to our knowledge) prior security reductions of the partition-based designs (e.g., [1,4,25,38,39]). Particularly, our reduction only blows up the running time by a small fixed polynomial (independent of ϵ), and does not lose an additional $O(\epsilon)$ in advantage, breaking the seemingly unavoidable tradeoff as above.

Scheme	# of ring vectors in the mpk	Bit length of id	RLWE Param $\frac{1}{\alpha} = \frac{q}{\sigma_{\mathsf{RLWE}}}$	# of ring vectors in $\mathsf{ct}/\mathsf{sk}_{\mathsf{id}}$	Reduction cost
[1]	$O(\lambda)$	$\Theta(\lambda)$	$\tilde{O}(n^{3.5})$	$O(1)$	$T' = T + \tilde{O}(\lambda^5 \cdot Q/\epsilon^2)$, $\epsilon' = O(\epsilon/(\lambda^5 Q))$
[25]	$O(\lambda^{\frac{1}{\mu}})^{\dagger}$	$\Theta(\lambda)$	$O(n^{0.5+2\mu})$	$O(1)$	$T' = O(T)$, $\epsilon' = O(((\frac{\lambda\epsilon}{Q})^{\mu}/\lambda)^{\mu+1})$
[39] I + [24]	$\omega(\log^2(\lambda))$	$\Theta(\lambda)$	$\tilde{O}(n^{5.5})$	$O(1)$	$T' = O(T)$, $\epsilon' = O(\epsilon^{v+1}/Q^v)$
[39] II	$\omega(\log(\lambda))$	$\Theta(\lambda)$	$\mathsf{poly}(n)^*$	$O(1)$	$T' = O(T)$, $\epsilon' = O(\epsilon^2/\lambda^2 Q)$
Ours A	$\omega(\log(\lambda))$	$\Theta(\lambda)$	$\tilde{O}(n^{4.5+\frac{4}{\kappa}})$	$O(1)$	$T' = T + \min\left\{\tilde{O}(\lambda^{1/\kappa} \cdot Q/\epsilon), O(\lambda^{(1+3/\kappa)} \cdot Q^{\kappa+3})\right\}$, $\epsilon' = O(\epsilon/\lambda^{1/\kappa} Q)^{\dagger\dagger}$
Ours B	$\omega(1)$	$\Theta(\lambda)$	$\tilde{O}(n^{7.5+\frac{4}{\kappa}})$	$O(1)$	$T' = T + \min\left\{\tilde{O}(\lambda^{1/\kappa} \cdot Q/\epsilon), O(\lambda^{(1+3/\kappa)} \cdot Q^{\kappa+3})\right\}$, $\epsilon' = O(\epsilon/\lambda^{1/\kappa} Q)^{\dagger\dagger}$

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

Notation: mpk, ct, and $\mathsf{sk}_{\mathsf{id}}$ denote the master public key, ciphertext, and secret key of the IBE. λ, n, q, σ_{RLWE} denote the security parameter, ring dimension, modulus, and gaussian parameter of RLWE. T, Q, and ϵ denote the adversary's running time, number of key queries and advantage in attacking the IBE scheme, and T', ϵ' denote the reduction's time and advantage in breaking RLWE. All the schemes have basic vector size of bit length $O(n \log^2 q)$. The size can be optimized to $O(n \log q)$ at the cost of increasing the size of q, which requires smaller RLWE parameter $1/\alpha$. All the schemes set the ring dimension $n = \Theta(\lambda)$. Here we use $\omega(f(\lambda))$ to denote any function that asymptotically dominates $f(\lambda)$, e.g., $\omega(1)$ can be $\log\log\lambda$ or $\log\log\log\lambda$, etc.

$*$ $\mathsf{poly}(n)$ 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 arbitrary. Since the reduction cost is exponential in μ, this value typically set very small (e.g., $\mu = 2$ or 3).

$\ddagger$ $v > 1$ is the constant that can be set small, depending on the underlying error correcting code.

$\dagger\dagger$ $\kappa \geq 1$ 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.

In Table 1, we summarize our results and a comparison with prior published works[6] in the asymptotic setting. To compare fairly with some prior schemes

[6] There is an unpublished work [4] that achieves essentially the same parameters as scheme II of [39], except [4] has an explicit bound on $q = O(n^{15.5})$.

described in the plain-LWE[7], we calculate the parameters of their ring variants and set the basic all vectors with the same bit length. We notice that parameters about some prior works in our table are different than the table of [39], which might over calculated some parameters. We also notice that there is a line of work, studying (almost) tightly secure IBE from lattices, e.g., [13, 26]. These constructions are not partition-based designs, and in general they require to homomorphically compute a PRF, resulting in at least $O(\lambda)$ basic ring vectors in mpk. In the context of "compact" IBE (for mpk), we believe that partitioned-based IBE are more suitable, so we only include these schemes for comparison.

Our scheme can be instantiated with multiple sets of parameters. We present two of them – scheme A requires smaller RLWE parameter $1/\alpha$, but require longer mpk, yet scheme B requires a slightly larger $1/\alpha$ but smaller mpk. Assuming that the security level of RLWE is roughly the same for any $1/\alpha = \mathsf{poly}(n)$, then scheme B would have smaller overall size, asymptotically.

Remark. We point out that it is possible to further shrink the mpk size of [39] I + [24] to $\omega(\log \lambda)$ basic ring vectors without applying the Barrington Theorem, by using an ECC with larger alphabets, e.g., [8], even though this idea was not explicitly written. This approach is similar to our scheme A, yet our scheme enjoys a tighter analysis. It is highly non-trivial to further shrink the mpk size to $\omega(1)$ ring vectors (as our scheme B), and this is the main novelty of this work.

A prior draft of this work would require to set $n = O(\lambda^{1+\tau})$ for a small constant τ, thus resulting in a larger length per basic ring vector. This work removes the requirement, showing that the typical setting $n = O(\lambda)$ is sufficient. As we mentioned in the note of Table 1, all the basic ring vectors (of all the listed schemes) have bit length $O(n \log^2 q) = O(\lambda \log^2 \lambda)$, which can be further optimized to $O(n \log q) = O(\lambda \log \lambda)$ by using a larger base of the gadget matrix. Thus, counting the number of ring vectors would be an easier way to compare efficiency/size of the listed schemes.

1.2 Technical Overview

We present an overview of our new techniques in two parts: (1) new IBE designs, and (2) tighter reduction analysis.

Part I: IBE Design

We start with a quick recap of some common features of the existing partitioned based IBE since [1], and then describe our new insights.

Recap of existing IBE designs At a high level, the public parameter of IBE [1] contains matrices $\mathbf{A}, \mathbf{B}_1, \ldots, \mathbf{B}_\ell$, where ℓ is the length of the identity, i.e.,

[7] The plain-LWE schemes usually count how many basic matrices in mpk, where each matrix is larger than the basic ring vectors of Ring-LWE designs by at least a multiplicative factor of $O(\lambda)$.

$\mathsf{id} \in \{0,1\}^\ell$. To derive a public key for an identity id, one just computes the matrix $\mathbf{F}_{\mathsf{id}} = [\mathbf{A}|\sum_{i\in[\ell]}(-1)^{\mathsf{id}[i]}\mathbf{B}_i]$. The encryption algorithm uses the dual-Regev scheme with respect to the matrix $\mathbf{F}_{\mathsf{id}}$. In the security proof, each public $\mathbf{B}_i$ is switched to $\mathbf{A}\cdot\mathbf{R}_i + h_i\mathbf{G}$ for some small-norm $\mathbf{R}_i$ and some random h_i. In this way, we can rewrite $\mathbf{F}_{\mathsf{id}} = [\mathbf{A}|\mathbf{A}\cdot\mathbf{R}_{\mathsf{id}} + H(\mathsf{id})\mathbf{G}]$, where $\mathbf{R}_{\mathsf{id}} = \sum_{i\in[\ell]}(-1)^{\mathsf{id}[i]}\mathbf{R}_i$, and $H(\mathsf{id}) = \sum_{i\in[\ell]}(-1)^{\mathsf{id}[i]}h_i$. The work [1] showed that suppose the hash function H isolates – with non-negligible probability, H separates the challenge id^* with the other query id's, then the scheme is adaptively secure. Later in subsequent works [4, 38, 39], it was observed that in fact we can view $\mathbf{B}_i$'s as GSW FHE ciphertexts [3, 21], and thus the key derivation process can be viewed as homomorphic computation of $H(\mathsf{id})$, (id in the clear and the description of H encrypted). Thus, by allowing the hash function to compute beyond the linear combination, it is possible to apply a more succinct hash function that can be encoded by much fewer public matrices.

Moreover, the work [25] showed that the plain LWE-based approach can be ported to the Ring-LWE setting (in 2-th powers cyclotomic rings), with a generic parameter saving. Particularly, the matrices can be replaced by ring vectors $\boldsymbol{a}, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_\ell$, and the intuition of homomorphic computation of the hash function works smoothly in the ring setting. Therefore, working in the ring has a generic advantage for smaller parameters than the plain LWE.

Challenges. Currently IBE with the shortest mpk (asymptotically) comes from the work [39], which proposed to use integer multiplication-then-modulo to design the hash function. Particularly, the hash function can be described by $a, b, \rho \in \mathbb{Z}$ such that $H_{a,b,\rho}(\mathsf{id}) = a \times \mathsf{id} + b \mod \rho$, where id is treated as an integer and the computation is in $\mathbb{Z}$. The work [39] showed that it suffices to encode $t = \omega(\log\lambda)$ bits of each a, b, ρ for the security analysis, and thus it suffices to use just ring vectors $\mathbf{b}_1, \ldots, \mathbf{b}_{3t}$ to encode the hash function, resulting in total $\omega(\log\lambda)$ matrices or ring vectors in the public parameter. Since integer multiplication-then-modulo is in NC1 [6], this homomorphic computation can be done within a polynomial modulus q by the Barrington's Theorem [23]. However, this approach does not give an explicit homomorphic computation method of the hash function, and it is hard to determine an explicit bound of q. This is one serious limitation of the current technique.

Our new insights. Our goal is to tackle the challenge as described above, and additionally, determine new methods to further shrink the size of mpk. To achieve this, we develop two new techniques: (1) a new homomorphic equality testing method under the Ring-LWE, and (2) a new family of hash functions in the ring setting that can be naturally computed homomorphically. By using these two techniques, we only need $\ell' = \omega(1)$ ring vectors in the mpk and our IBE design can be computed explicitly without the Barrington's Theorem.

New Technique (1): As we discussed above, the design of IBE is highly related to homomorphic computation of a hash function. To shrink the size of mpk, it suffices to construct a more efficient GSW style encoding that can pack/unpack

multiple bit encodings into one encoding. We then observe that this task is deeply connected to the homomorphic equality test as we elaborate how next.

The most general form of the homomorphic equality test is given an encoding $\mathsf{Encode}(\alpha)$ and some β in the clear, homomorphically compute an encoded bit $\mathsf{Encode}(\tau)$ such that $\tau = 1$ if and only if $\alpha = \beta$. Denote this family of functions as $\{\mathsf{Equal}_\beta(\alpha)\}$ where each function is parameterized by β in the clear. If we can achieve this task beyond bit compute, i.e., α can be some ring element, then we can homomorphically extract every single bit of α from $\mathsf{Encode}(\alpha)$ by the equality test, by computing $\sum_{\beta \in Z} \mathsf{Equal}_\beta(\alpha)$ where Z is the set of all possible values that have consistent bit with α for the targeted bit we want to extract. (We present the detailed procedure in Section 3). However, the general task seems to incur a large blowup in the noise, and thus unclear whether it is feasible.

This work identifies a critical property of cyclotomic rings so that we can achieve an important subclass of the task. Particularly, let us take R as the m-th cyclotomic ring where m is a power of two. In this case, we know that $R = \mathbb{Z}[x]/(x^n + 1)$ where $n = \varphi(m) = m/2$. Then, we consider the case where α appears in the exponent of the monomial x (corresponding to a root of unity in cyclotomic rings); i.e., given $\mathsf{Encode}(x^\alpha)$ and $\beta \in \mathbb{Z}$, compute the desired $\mathsf{Encode}(\tau)$. To design a homomorphic equality test function in this ring setting, we first observe a critical fact in the rings. For any monomial $v = x^i$ where $i \neq 0 \mod m$, we have $f(v) := \sum_{i=0}^{m-1} v^i = \frac{1-v^m}{1-v} = 0$, as the denominator $1 - v$ is not equal to 0, and $1 - v^m = 1 - x^{mi} = 0$ for $i \neq 0 \mod m$. On the other hand, if $v = 1$, i.e., $i = 0 \mod m$, then $f(v) = m$. Therefore, the function f naturally separates the two cases as: $f(x^i) = \begin{cases} 0 & \text{if } i \neq 0 \mod m \\ m & \text{otherwise.} \end{cases}$

Using this fact, we can design a simple algorithm for our goal: given $\boldsymbol{b} = \mathsf{Encode}(x^\alpha)$ and β, we compute the following three steps: (1) first we set $\boldsymbol{b}' = \mathsf{Encode}(x^{\alpha-\beta})$ by a homomorphic scale multiplication of $x^{-\beta}$.

(2) Then we homomorphically compute $\boldsymbol{b}'' = f(\boldsymbol{b}') = \mathsf{Encode}(f(x^{\alpha-\beta}))$. (3) Finally, we output $\boldsymbol{b}^*$ by homomorphically multiply $\boldsymbol{b}''$ and $\mathsf{Encode}(m^{-1})$.[8] Clearly, this procedure outputs $\mathsf{Encode}(\tau)$ where $\tau = 1$ if $\alpha = \beta \mod m$ and otherwise 0. Our analysis crucially relies on that multiplying monomials does not blow up the norm of a matrix, and thus the noise behaves the same as the bit multiplication case.

By using the above techniques, we can pack/unpack $\log(m)$ bit encodings into one single encoding. This would imply that we can further shrink the size of mpk required in the work [39] by a factor of $O(\log m)$, resulting $t = \omega(1)$ ring vectors for mpk under the Ring-LWE setting. This algebraic structure of Ring-LWE demonstrates another non-trivial efficiency gain over the plain LWE, which may be of independent interests.

New Technique (2): By building upon the equality test technique, we further design a new hash function that can be explicitly computed, homomorphically, without the Barrington's Theorem. We start with a nice observation by [4] that

[8] We note that m^{-1} with respect to $\mathbb{Z}_q$ exists if we choose m and q to be co-prime.

identifies that in fact (almost) pairwise independent hash functions suffice to isolate [35]. To design a suitable hash function, we propose to use an error-correcting code ($\mathsf{ECC} : \{0,1\}^{\ell} \to \mathbb{Z}_p^{\mathsf{L}}$) with good relative distance. We first consider the hash function $H_{\alpha,\beta}(\mathsf{id}) = \mathsf{ECC}(\mathsf{id})[\alpha] + \beta$. It is not hard to show that this hash function behaves as an almost pairwise independent hash. The drawback is that the range might be too small in the application of IBE designs. To amplify the range, we can use a parallel repetition: $H^{\parallel}_{\boldsymbol{\alpha},\boldsymbol{\beta}}(z) = \left(H_{\boldsymbol{\alpha}_1,\boldsymbol{\beta}_1}(z), \ldots, H_{\boldsymbol{\alpha}_t,\boldsymbol{\beta}_t}(z)\right)$. In fact, using error correcting codes to design a partition function has been explored in the context of IBE and VRF, e.g., [8, 24, 39], yet these generic designs are still not naturally compatible with the ring setting. Our new insight is to design an embedding method that maps the output $H^{\parallel}_{\boldsymbol{\alpha},\boldsymbol{\beta}} \in \mathbb{Z}_p^t$ to the ring R of the underlying Ring-LWE. In this way, the homomorphic computation method can be designed based on the above equality test method, and therefore our IBE design can be explicit, avoiding the route of the Barrington's Theorem. The actual design requires to deal with further technical subtleties. We refer the readers to Section 4 for details.

Part II: Tighter Reduction Analysis

Next we present our new insights to achieve a tighter analysis for general partitioned-based IBE designs. We start with a recap of the existing proof framework.

Recap of the proof framework. As we discussed above, the security proof framework switches the public matrices (or ring vectors) $\mathbf{B}_i$ to $\mathbf{A} \cdot \mathbf{R}_i + h_i \mathbf{G}$, and then homomorphically computes $\mathbf{F}_{\mathsf{id}} = [\mathbf{A} | \mathbf{A} \cdot \mathbf{R}_{\mathsf{id}} + H(\mathsf{id})\mathbf{G}]$, for some suitable hash function H. Intuitively, the security reduction can respond to a key query id if $H(\mathsf{id}) \neq 0$, and then embeds the (Ring) LWE challenge if $H(\mathsf{id}^*) = 0$ for the challenge id^*. Therefore, if the hash function separates all the query id's from the challenge id^* as we just stated, then the reduction can be used to attack the underlying (Ring) LWE. On the other hand, if the adversary queries *some* id that $H(\mathsf{id}) = 0$, then the reduction simply aborts and outputs a random guess. By designing an appropriate parameters for H, we can show that with some noticeable probability, we will have $H(\mathsf{id}) \neq 0$ for all id's queried by the adversary and $H(\mathsf{id}^*) = 0$. This implies that the security reduction will still have sufficient advantage in attacking the underlying (Ring) LWE.

Challenges. To analyze the limitation of the current reduction approach, we delve into some further details. First we denote as the event abort if the adversary has queried some id such that $H(\mathsf{id}) = 0$ or $H(\mathsf{id}^*) \neq 0$, and $\neg\mathsf{abort}$ as the other case. Let $\gamma(I)$ denote the probability of $\neg\mathsf{abort}$ for the query pattern $I = \{\mathsf{id}_1, \ldots, \mathsf{id}_t\}$ for some $t \leq Q$, and $\gamma(I) \in [\gamma_{\min}, \gamma_{\max}]$ for every query pattern I.

The work [1, 7, 37] showed the following statement (simplified): suppose the adversary has advantage ϵ in breaking the IBE scheme, then by this partitioning strategy, the reduction would have advantage roughly $\epsilon\gamma_{\min} - (\gamma_{\max} - \gamma_{\min})/2$ in breaking the (Ring) LWE hard problem. Now (also pointed out by [1]), we would face a challenge in choosing the range $[\gamma_{\min}, \gamma_{\max}]$ (by setting appropriate the hash function parameters):

– If we aim to optimize the reduction's advantage, we can set $\gamma_{\max} \approx 1/Q$ and $\gamma_{\min} \approx 1/2Q$. However, as $\epsilon\gamma_{\min}$ might be smaller than the extra term $(\gamma_{\max} - \gamma_{\min})/2$, we need to apply the technique of Waters [37] that reduces the gap between $\gamma_{\min}$ and $\gamma_{\max}$ by adding an extra "artificial abort". However, this would require to blow up the running time by roughly $O(1/\epsilon^2)$.
– The other way to handle this is by Bellare and Ristenpart [7], which is then used by the follow up works [4, 25, 38, 39]. Particularly, they choose $\gamma_{\max} \approx \epsilon/Q$ and $\gamma_{\min} \approx \epsilon(1-\epsilon)/Q$, so that the gap would be ϵ^2/Q, implying $\epsilon\gamma_{\min} - (\gamma_{\max} - \gamma_{\min})/2 \geq \epsilon^2/2Q$. This does not need to blow up the running time, yet the advantage would suffer from an extra multiplicative loss of ϵ compared with the above.

Our new insights. To break the tradeoff, we first give a new method that can generally improve both of the above two cases: for the former, the running time blowup is improved to $O(1/\epsilon)$, and for the latter, the advantage only loses an extra multiplicative $\sqrt{\epsilon}$. Then we show how to further reduce the running time blowup for the first case, so that it can be upper bounded by a fixed polynomial (in n, Q) without relying on the advantage ϵ. The crux for the first idea relies on using the framework [30], on which we devise a better advantage bound than that of $\epsilon\gamma_{\min} - (\gamma_{\max} - \gamma_{\min})/2$. The second idea uses a critical property of the design of the hash function. We elaborate the insights below.

First we recall the work [30], which considers two quantities α, β, where the former is the probability that an adversary does not output $\perp$, and the latter is the conditional probability that the adversary outputs the correct bit, conditioned on the non-$\perp$ event. Then the work [30] defined the advantage in a decisional game $\epsilon := \alpha(1-2\beta)^2 = \alpha\delta^2$ where $\delta = |1-2\beta|$.

Now we analyze the reduction above under this framework. Consider an (α, β) adversary with advantage $\epsilon = \alpha\delta^2$. If we take the reduction as above, then the reduction has $\gamma_{\min}$ probability of $\neg\mathsf{abort}$, resulting in non-$\perp$ probability $\alpha' = \alpha\gamma_{\min}$ as the hash is chosen independent of the adversary. By a careful analysis, the reduction's conditional success probability would be roughly $\beta' \approx (\gamma_{\min}/\gamma_{\max}) \cdot \beta$. In order to ensure a significant success (conditional) probability of the reduction, i.e., sufficiently large $\delta' = |1-2\beta'|$, we aim to set $\gamma_{\min}/\gamma_{\max} \approx 1 - \delta/4$, meaning that $\delta' = |1-2\beta'| \approx |1-2(1-\delta/4)\beta| = |\pm\delta + \delta\beta/2| \geq \delta/2 \geq \sqrt{\varepsilon}/2$. Now the reduction has advantage $\alpha'\delta'^2 \approx \alpha \cdot \gamma_{\min} \cdot \delta^2/4 \approx \epsilon \cdot \gamma_{\min}/4$.

Now we can improve the parameters with or without the artificial abort:

– We can improve the running time of the first case compared with the previous analysis. Particularly, we set $\gamma_{\max} \approx 1/Q$ and $\gamma_{\min} \approx 1/2Q$. The ratio of $\gamma_{\min}/\gamma_{\max}$ is 0.5, which needs to be increased to $(1-\delta/4)$ by the artificial abort technique of Waters [37]. Yet now, we only need precision $O(\delta) = O(\sqrt{\epsilon})$, which yield $O(1/\epsilon)$ samples, whereas the prior analysis needs precision $O(\epsilon)$, and thus $O(1/\epsilon^2)$ samples.
– We can also improve the reduction's advantage for the second case. Particularly, we can set $\gamma_{\max} \approx \delta/4Q$ and $\gamma_{\min} \approx \delta(1-\delta/4)/4Q$. In this way, the ratio is $(1-\delta/4)$ as needed, and the reduction's advantage would lose

a multiplicative factor of $O(\delta) = O(\sqrt{\epsilon})$ compared with the above, whereas the prior analysis would lose $O(\epsilon)$.

Finally, we show how to further improve the reduction's running time for the first case, to get rid of the dependency on $O(1/\epsilon)$, which would be large when ϵ is small. As a result, our reduction has a smaller overhead in running time, i.e., $T + \mathsf{poly}(\lambda)$ for some small polynomial that is independent of ε (recall that T is adversary's running time), and maintains the advantage, achieving the best of the both of the two cases.

To achieve this, we observe that the blowup in running time comes from the estimation of $\gamma(I)$ for the technique of Waters' artificial abort, which roughly needs $O(1/\epsilon)$ samples for the procedure. To get rid of this dependency, we observe that the sample space of the our design of hash function H (all possible choices of the hash function) is roughly bounded by a small fixed polynomial $\mathsf{poly}(\lambda)$. Therefore, if the adversary has a larger advantage ϵ, then the reduction would use $O(1/\epsilon)$ samples to estimate $\gamma(I)$, whereas if the ϵ is small, then the reduction would enumerate all possible choices of the hash function to compute the *exact value* of $\gamma(I)$. Therefore, the running time in the worst case would be upper bounded by $T + \mathsf{poly}(\lambda)$ as desired.

2 Preliminaries

This section includes the basic preliminaries. Readers who are already familiar with the concepts can skip the entire section and start to read from Section 3.

Notations. We denote $\mathbb{Z}$ as the set of the integers and $\mathbb{R}$ as the real numbers. For a positive integer k, let $[k]$ be set of integers $\{0, 1, ..., k-1\}$. We denote $[a, b]$ as the set $[a, b] \cap \mathbb{Z}$ for any integers $a, b \in \mathbb{N}$ satisfying $a \leq b$. We use bold uppercase letters to denote matrices (e.g., $\mathbf{A}$), and bold lowercase letters for column vectors (e.g., $\boldsymbol{a}$), and denote the horizontal concatenation of two vectors $\boldsymbol{a}, \boldsymbol{b}$ by $[\boldsymbol{a}|\boldsymbol{b}]$. For any $1 \leq p \leq \infty$, the p-norm of a vector $\boldsymbol{a}$ is defined as $\|\boldsymbol{a}\|_p = (\sum_i \|\boldsymbol{a}_i\|^p)^{1/p}$, and p-norm of a matrix $\mathbf{A}$ is defined by $\|\mathbf{A}\|_p = \max_{\|\boldsymbol{x}\|_p=1} \|\mathbf{A}\boldsymbol{x}\|_p$, assuming the dimensions match. We omit the subscript p if $p = 2$. We denote $s_1(\mathbf{A})$ as the largest singular value of $\mathbf{A}$, then we have $s_1(\mathbf{A}) = \|\mathbf{A}\|$. We say $\varepsilon : \mathbb{N} \to [0, 1)$ be a negligible function, if for any $c > 0$, we have $\varepsilon(n) < \frac{1}{n^c}$ starting from some integer $n_0(c) \in \mathbb{N}$. We say an event happens with overwhelmingly, if the probability of that event not happens is negligible. For any two random variables X and Y with support Ω, define the statistical distance, denoted $\Delta(X, Y)$, as $\Delta(X, Y) = \frac{1}{2} \sum_{s \in \Omega} |\mathsf{Pr}[X = s] - \mathsf{Pr}[Y = s]|$. We say X is statistically close(or ϵ-close) to Y, if the statistical distance $\Delta(X, Y)$ is negligible(or $\Delta(X, Y) \leq \epsilon$).

Definition 2.1 (Relative Distance) *Let $\mathbb{F}$ be some finite field and $\mathsf{L} \in \mathbb{N}$, $\mathcal{D}$ be some input domain, and $\mathsf{ECC} : \mathcal{D} \to \mathbb{F}^{\mathsf{L}}$ be some encoding, where the output vector is indexed by $[1, \ldots, \mathsf{L}]$. Define the relative distance of ECC, denoted Υ, as*

$$\Upsilon := \min \left\{ \mathsf{Pr}_{i \xleftarrow{\$} [1,\ldots,\mathsf{L}]} \left[\mathsf{ECC}(\boldsymbol{a})[i] \neq \mathsf{ECC}(\boldsymbol{b})[i] \right] \Big| \boldsymbol{a} \neq \boldsymbol{b}, \boldsymbol{a}, \boldsymbol{b} \in \mathcal{D} \right\}$$

2.1 Identity-Based Encryption (IBE)

Definition 2.2 (IBE [11, 33]) *An identity-based encryption scheme Π consists of four algorithms* $\{\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}\}$ *as follows.*

- $\mathsf{Setup}\ (1^\lambda)$*: On input the security parameter* λ*, the algorithm outputs the master public key* mpk *and the master secret key* msk*.*
- $\mathsf{KeyGen}\ (\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$*: On input* $(\mathsf{mpk}, \mathsf{msk})$ *and an identity* id*, the key generation algorithm outputs a secret key* $\mathsf{sk}_{\mathsf{id}}$ *corresponding to the identity* id*.*
- $\mathsf{Enc}\ (\mathsf{mpk}, \mathsf{id}, \mu)$*: On input the master public key* mpk*, identity* id *and the message* μ*, the encryption algorithm outputs a ciphertext* ct*.*
- $\mathsf{Dec}\ (\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}}, \mathsf{ct})$*: On input the master public key* mpk*, the secret key* $\mathsf{sk}_{\mathsf{id}}$ *and the ciphertext* ct*, the decryption algorithm outputs the message* μ' *or* $\perp$*.*

Correctness. We say an IBE scheme Π is correct, if for any message μ and any identity id, the following holds

$$\mathsf{Pr}\left[\mathsf{Dec}(\mathsf{sk}_{\mathsf{id}}, \mathsf{ct}) \neq \mu \;\middle|\; \begin{array}{l}(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id}) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, \mu)\end{array}\right] < \mathsf{negl}(\lambda).$$

Security. We use the following experiment to describe the security of IBE against adaptive adversaries. Formally, for any PPT adversary $\mathcal{A}$, we consider the experiment $\mathbf{Expt}^{\mathsf{IBE}}_{\mathcal{A}}(1^\lambda)$ between $\mathcal{A}$ and the challenger defined below:

Setup: At the beginning of the experiment, the adversary $\mathcal{A}$ sends a public parameter requirement to the challenger. After receiving the public parameter requirement, the challenger runs $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$, and sends mpk to the adversary $\mathcal{A}$.

Phase 1: Proceeding adaptively, the adversary $\mathcal{A}$ queries a sequence of identities $(\mathsf{id}_1, \cdots, \mathsf{id}_m)$. On the i-th query, the challenger runs $\mathsf{KeyGen}(\mathsf{msk}, \mathsf{id}_i)$, and sends the result $\mathsf{sk}_{\mathsf{id}_i}$ to the $\mathcal{A}$.

$\mathsf{Challenge}$: In this phase, $\mathcal{A}$ chooses an identity $\mathsf{id}^* \notin \{\mathsf{id}_1, \cdots, \mathsf{id}_m\}$ and two length-equal messages μ_0, μ_1, and forwards them to the challenger. Upon receiving the $\mathsf{id}^*, \mu_0, \mu_1$, the challenger chooses a random bit $b \in \{0, 1\}$ and runs $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{id}^*, \mu_b)$. Then, the challenger sends ct^* to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ continues to make key queries $(\mathsf{id}_{m+1}, \cdots, \mathsf{id}_Q)$ such that $\mathsf{id}_j \neq \mathsf{id}^*$ for any $j \in [m+1, Q]$. The challenger responds as in Phase 1.

Guess: The adversary $\mathcal{A}$ outputs a bit b' as the guess of b.

We define the notion of asymptotic security: the IBE scheme is secure if for any PPT adversary $\mathcal{A}$, the probability that $\mathcal{A}$ outputs the right bit, i.e., $b' = b$ in $\mathbf{Expt}^{\mathsf{IBE}}_{\mathcal{A}}(1^\lambda)$ is bounded by $\frac{1}{2} + \mathsf{negl}(\lambda)$ for some negligible function $\mathsf{negl}(\lambda)$.

In addition to the asymptotic notion, our work also focuses on the concrete bit-security notion, which is more relevant in practice. In the following section, we present the framework established by the recent work [30].

2.2 Concrete Bit-security

The work [30] considers concrete bit-security for security games that capture two types of general primitives – (1) search primitives where the adversary's goal is to output a string that satisfies a certain relation, and (2) decision primitives where the adversary only needs to output one bit, trying to distinguish two challenging distributions. Clearly, IBE is a decision primitive as the adversary in $\mathbf{Expt}_{\mathcal{A}}^{\mathsf{IBE}}(1^\lambda)$ above tries to guess the challenge bit. To capture its bit-security, we present the framework of [30] for decision primitives.

Given an adversary $\mathcal{A}$, we say $\mathcal{A}$ is a $(T_{\mathcal{A}}, \alpha_{\mathcal{A}}, \beta_{\mathcal{A}})$-adversary if its running time is at most $T_{\mathcal{A}}$, output probability $\alpha_{\mathcal{A}} = \mathsf{Pr}[\mathcal{A} \neq \perp]$, and conditional success probability $\beta_{\mathcal{A}} = \mathsf{Pr}[\mathcal{A} \text{ wins} \mid \mathcal{A} \neq \perp]$, where the probabilities are over the randomness of the entire game. For a decision primitive including IBE, define the advantage of the $(T_{\mathcal{A}}, \alpha_{\mathcal{A}}, \beta_{\mathcal{A}})$-adversary $\mathcal{A}$ as $\mathsf{Adv}_{\mathcal{A}} := \alpha_{\mathcal{A}}(2\beta_{\mathcal{A}} - 1)^2$.

Importantly, this formulation allows the adversary to output $\perp$, intuitively meaning "I don't know" even for decision primitives. As the work [30] showed, in some cases it is more advantageous if the adversary admits being defeated rather than guessing at random. In this work, we demonstrate that this is extremely crucial for partition-based IBE [1,7,37,39], allowing a much better security analysis over all these prior work. For the rationale of this definitional framework, we refer the reader to the original paper [30]. Next we present the notion of bit-security for IBE in the framework of [30] as a general decision primitive.

Definition 2.3 ([30]) *We say an* IBE *scheme is adaptively secure with λ-bit security, if for all (T, α, β)-adversary $\mathcal{A}$ in $\mathbf{Expt}_{\mathcal{A}}^{\mathsf{IBE}}(1^\lambda)$, we have $\frac{T}{\mathsf{Adv}_{\mathcal{A}}} \geq 2^\lambda$.*

Remark 2.4 *The term $T_{\mathcal{A}}$ can also be generalized to any measure of resources that is linear under repetition as stated in [30]. In this work, we use the running time for simplicity. Moreover, we assume that $T_{\mathcal{A}}$ is greater than the running time of the challenger. This is without loss of generality as the security game ends at the last guessing step of the adversary, whose total running time must be at least as long as that of experiment (including the challenger's time).*

Next we present a useful lemma for the relation between the statistical distance between two games and the difference of the corresponding (α, β)'s. Due to space limit, we put the proof of the lemma below in full version of our paper.

Lemma 2.5 *Let $\mathcal{S}^{\mathcal{P}}, \mathcal{S}^{\mathcal{Q}}$ be two indistinguishability games with black-box access to two probability distribution $\mathcal{P}$ and $\mathcal{Q}$, respectively, with $\Delta(\mathcal{P}, \mathcal{Q}) \leq \varepsilon$. For any $(T_{\mathcal{A}}, \alpha_{\mathcal{A}}^{\mathcal{P}}, \beta_{\mathcal{A}}^{\mathcal{P}})$-adversary $\mathcal{A}$ with $\alpha_{\mathcal{A}}^{\mathcal{P}} > \varepsilon$ in the game $\mathcal{S}^{\mathcal{P}}$, the same $\mathcal{A}$ in the game $\mathcal{S}^{\mathcal{Q}}$ is a $(T_{\mathcal{A}}, \alpha_{\mathcal{A}}^{\mathcal{Q}}, \beta_{\mathcal{A}}^{\mathcal{Q}})$-adversary, where $\alpha_{\mathcal{A}}^{\mathcal{Q}} \geq \alpha_{\mathcal{A}}^{\mathcal{P}} - \varepsilon$ and $\beta_{\mathcal{A}}^{\mathcal{Q}} \geq \beta_{\mathcal{A}}^{\mathcal{P}} - \varepsilon/(\alpha_{\mathcal{A}}^{\mathcal{P}} - \varepsilon)$.*

2.3 Lattices and Gaussian Distributions

Lattices. A lattice is a discrete additive subgroup of $\mathbb{R}^n$. Let $\mathbf{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m) \subset \mathbb{R}^{n \times m}$ consist of m linearly independent vectors, the n-dimensional lattice Λ generated by the basis $\mathbf{B}$ is $\Lambda = \mathcal{L}(\mathbf{B}) = \{\mathbf{B} \cdot \boldsymbol{c} = \sum_{i \in [m]} c_i \cdot \boldsymbol{b}_i : \boldsymbol{c} = (c_0, \ldots, c_{m-1}) \in$

$\mathbb{Z}^m\}$. We denote $\widetilde{\mathbf{B}}$ as the Gram-Schmidt orthogonalization of $\mathbf{B}$, and $\|\mathbf{B}\|_{\mathsf{GS}}$ as the length of the longest vector of $\widetilde{\mathbf{B}}$.

In this paper, we focus on a particular family of integer lattices. Let $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ for integers m, n, q, where m and q are functions of n. We consider the following two kinds of full-rank m-dimensional integer lattices defined by $\Lambda_q^{\perp}(\mathbf{A}) = \{\boldsymbol{e} \in \mathbb{Z}^m : \mathbf{A}^\top \cdot \boldsymbol{e} = 0 \bmod q\}$ and its shift $\Lambda_q^{\boldsymbol{u}}(\mathbf{A}) = \{\boldsymbol{e} \in \mathbb{Z}^m : \mathbf{A}^\top \cdot \boldsymbol{e} = \boldsymbol{u} \bmod q\}$.

Gaussian Distributions. For any real number $s > 0$ and an n-dimensional vector $\boldsymbol{c}$, let $\rho_{s,\boldsymbol{c}}(\boldsymbol{x}) := \exp(-\pi\|\boldsymbol{x}-\boldsymbol{c}\|^2/s^2)$ be the gaussian function with parameter s and centered at $\boldsymbol{c}$. The discrete gaussian distribution over a lattice coset $\Lambda + \boldsymbol{u}$ is defined as $D_{\Lambda+\boldsymbol{u},s}(\boldsymbol{x}) = \frac{\rho_s(\boldsymbol{x})}{\rho_s(\Lambda+\boldsymbol{u})}$. Let $\eta_{\epsilon_s}(\Lambda)$ be the smoothing parameter. For a gaussian over lattices, we have the following tail bound.

Lemma 2.6 ([20, 29]) *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 $\boldsymbol{c} \in \mathrm{span}(\Lambda)$, we have $\mathsf{Pr}\big[\|D_{\Lambda+\boldsymbol{c},s}\| \geq s\sqrt{n}\big] \leq 2^{-n} \cdot \frac{1+\epsilon_s}{1-\epsilon_s}$. Furthermore, if $\boldsymbol{c} = 0$, the bound holds for any $r > 0$ with $\epsilon_s = 0$.*

We say a polynomial $a = \sum_{i\in[n]} a_i x^i$ is sampled from gaussian distribution $D_{\Lambda+\boldsymbol{u},s}$, if the coefficient vector $(a_0, ..., a_{n-1})$ is sampled by $D_{\Lambda+\boldsymbol{u},s}$. We further define the gaussian distribution $D_{\Lambda+\boldsymbol{u}}^{\mathsf{Coeffs}}$ as the distribution of a polynomial $a = \sum_{i\in[n]} a_i x^i$ sampled from gaussian distribution $D_{\Lambda+\boldsymbol{u},s}$ .We also extend this notion to the polynomial vector $\boldsymbol{a} = (a_1, \cdots, a_n)$ component-wise.

Sub-Gaussian. It is convenient for our analyses to use sub-Gaussian random variables and their bounds. We defer the details to full version of this paper.

2.4 Rings and Ideal Lattices

Next, we briefly present the concepts and lemmas related to rings and ideal lattices required in this work. See the work of [27, 28] for further details.

Rings. For an m-th cyclotomic polynomial $\Phi(x)$ (of degree $n = \varphi(m)$), define the polynomial quotient ring $R = \mathbb{Z}[x]/\Phi(x)$. For an integer q, denote R_q as the ring R/qR. For the polynomial ring R, we denote $[-\rho, \rho]_R \subset R$ as the set of elements in R with coefficients in $[-\rho, \rho]\cap\mathbb{Z}$. Any element in R can be considered as a vector of its coefficients. Namely, an element $a = \sum_{i\in[n]} a_i x^i \in R$ can be seen as the vector $\boldsymbol{a} = (a_0, ..., a_{n-1})$. We call this map as coefficient embedding (denoted as $\mathsf{Coeffs}(\cdot)$). Furthermore, we can also represent a ring element $a \in R$ as a matrix in $\mathbb{Z}^{n\times n}$ by the following map $\mathsf{Rot} : R \to \mathbb{Z}^{n\times n}$:

$$\mathsf{Rot}(a) = \begin{bmatrix} \mathsf{Coeffs}(a)^\top \\ \mathsf{Coeffs}(xa \bmod \Phi(x))^\top \\ \vdots \\ \mathsf{Coeffs}(x^{n-1}a \bmod \Phi(x)\ )^\top \end{bmatrix}.$$

Furthermore, we extend this map to ring vectors and matrices by applying it entry-wise, i.e., for a vector $\boldsymbol{a}^\top = (a_1, \ldots, a_m) \in R^m$, we define $\mathsf{Rot}(\boldsymbol{a}^\top) =$

$[\mathsf{Rot}(a_1)|\dots|\mathsf{Rot}(a_m)] \in \mathbb{Z}^{n\times nm}$, and the map for matrices can be defined similarly. In the case of power of 2 cyclotomic rings, i.e., $\Phi(x) = x^n + 1$ for n being some power of 2, the above rotation matrix $\mathsf{Rot}(a)$ is the anti-cyclic matrix.

Rings in This Work. Throughout this paper, we only work on power of 2 cyclotomic rings for their nice and simple mathematical structures. Thus, we will only present the related lemmas with respect to this type of rings.

Norms and Singular Value. The norms of ring vectors (or matrices) are defined by their corresponding coefficient embedding vectors (or matrices). The singular value of a ring matrix $\mathbf{R} \in R^{k\times k'}$ is defined by the singular value of its corresponding matrix obtained by Rot map, that is $s_1(\mathbf{R}) := \sup_{\|\boldsymbol{u}\|=1} \|\mathsf{Rot}(\mathbf{R})\boldsymbol{u}\|$.

The following lemma shows that R_q has exponentially many invertible elements, if the modulo q satisfies certain property.

Lemma 2.7 ([25]) *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 $u \in R_q$ satisfying $\|\mathsf{Coeffs}(u)\|_2 < \sqrt{q}$ are invertible, i.e., $u \in R_q^*$.*

Ring Learning with Errors. The Learning With Errors (LWE) problem was introduced by Regev [32]. To improve efficiency of LWE-based schemes, the ring version of LWE, namely RLWE, was introduced [27, 34]. For $s \in R_q$ and an error distribution ψ over R_q, the RLWE distribution $A_{s,\psi}$ over $R_q \times R_q$ is the distribution of the pair $(a, b = (a \cdot s) + e)$, where a is randomly sampled over R_q, and the error term e is independently sampled according ψ. Here we recall the RLWE problem as follows.

Definition 2.8 (Decision Ring-LWE Problem) *The decision Ring-LWE problem, denoted R-$\mathsf{DLWE}_{n,\ell,q,\psi}$ is to distinguish between ℓ independent samples from $A_{s,\psi}$ for a random choice of a secret $s \leftarrow R_q$ of degree n, and the same number of uniformly random and independent samples from $R_q \times R_q$.*

The bit hardness can be defined following the framework [30] as a decision primitive, similar to the case of IBE in Definition 2.3. Particularly, the R-$\mathsf{DLWE}_{n,\ell,q,\psi}$ problem can be formulated by a security game $\mathbf{Expt}_{\mathcal{B}}^{\mathsf{RLWE}}(1^n, \ell, q, \psi)$ where an adversary $\mathcal{B}$ is challenged with either ℓ samples from $A_{s,\psi}$ or the uniform distribution. Define $\mathsf{Adv}_{\mathcal{B}}^{\mathsf{RLWE}} = \alpha_{\mathcal{B}} \cdot (2\beta_{\mathcal{B}} - 1)^2$, where $\alpha_{\mathcal{B}}$ and $\beta_{\mathcal{B}}$ are the probability that $\mathcal{B}$ does not abort and the conditional probability that $\mathcal{B}$ outputs the correct bit conditioning on the non-abort event. Then the bit hardness of R-DLWE is defined as follows.

Definition 2.9 (Bit Hardness of R-DLWE) *R-$\mathsf{DLWE}_{n,\ell,q,\psi}$ is λ-bit hard, if for all (T, α, β)-adversary $\mathcal{B}$ in $\mathbf{Expt}_{\mathcal{B}}^{\mathsf{RLWE}}(1^n, \ell, q, \psi)$, we have $\frac{T}{\mathsf{Adv}_{\mathcal{B}}^{\mathsf{RLWE}}} \geq 2^\lambda$.*

Below we present a reduction from some lattice problem to R-DLWE, showing that the ring (D)LWE problem is as hard as the underlying lattice problem.

Lemma 2.10 (Theorem 1 of [25]) *Let α be the positive real, m be a power of 2, ℓ be an integer, $\Phi(x) = x^n + 1$ be the mth 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 $\sigma_{\mathsf{RLWE}} := \alpha q \geq n^{3/2}\ell^{1/4}\omega(\log^{9/4}(n))$. Then, there is a* PPT *quantum reduction from $\tilde{O}(n/\alpha)$-approximate* SIVP*(or* SVP*) to R-$\mathsf{DLWE}_{n,\ell,q,\chi}$ with $\chi = D^{\mathsf{Coeffs}}_{\mathbb{Z}^n,\sigma_{\mathsf{RLWE}}}$.*

Trapdoors for Rings. For positive integers b and $k > k' \geq \lceil \log(q) \rfloor$, let $\boldsymbol{g}_b^\top = [1|b|b^2|...|b^{k'}|\mathbf{0}] \in R^k$ be the gadget matrix. As stated in the work of [29], this gadget matrix has a public trapdoor $\mathbf{T}_{\boldsymbol{g}}$ with small norm, i.e., $\|\mathbf{T}_{\boldsymbol{g}}\| \leq \sqrt{b^2+1}$. Next we present several useful sampling algorithms from the work of [25, 29].

Lemma 2.11 ([25]) *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:*

- $\mathsf{TrapGen}(n,k,\rho,q) \to (\boldsymbol{a}, \mathbf{T}_{\boldsymbol{a}})$ ([29], Lemma 5.3): *A randomized algorithm that, when $k \geq 2\log_\rho(q)$, outputs a ring vector $\boldsymbol{a} \in R^k$ and a matrix $\mathbf{T}_{\boldsymbol{a}} \in R^{k\times k}$, where $\mathsf{Rot}(\boldsymbol{a}^\top) \in \mathbb{Z}^{n\times nk}$ is full-rank matrix and $\mathsf{Rot}(\mathbf{T}_{\boldsymbol{a}}) \in \mathbb{Z}^{nk\times nk}$ is a basis for $\Lambda^\perp((\mathsf{Rot}(\boldsymbol{a}^\top))$ such that $\boldsymbol{a}$ is $\frac{k}{2^{\frac{nk}{4}+2}}$-close to uniform and $\|\mathsf{Rot}(\mathbf{T}_{\boldsymbol{a}})\|_{\mathsf{GS}} < O\left(b\rho\sqrt{n\log_\rho(q)}\right)$.*
- $\mathsf{SampleLeft}(\boldsymbol{a},\boldsymbol{b},\mathbf{T}_{\boldsymbol{a}},u,\sigma) \to \boldsymbol{e}$ ([16]): *A randomized algorithm that, on input the vectors $\boldsymbol{a},\boldsymbol{b} \in R^k$, where $\mathsf{Rot}(\boldsymbol{a}^\top), \mathsf{Rot}(\boldsymbol{b}^\top) \in \mathbb{Z}^{n\times nk}$ are full-rank, an element $u \in R_q$, a matrix $\mathbf{T}_{\boldsymbol{a}}$ such that $\mathsf{Rot}(\mathbf{T}_{\boldsymbol{a}}) \in \mathbb{Z}^{nk\times nk}$ is a basis for $\Lambda^\perp((\mathsf{Rot}(\boldsymbol{a}^\top))$, and a Gaussian parameter $\sigma > \|\mathsf{Rot}(\mathbf{T}_{\boldsymbol{a}})\|_{\mathsf{GS}} \cdot \sqrt{\log(2n(1+1/\epsilon_s))/\pi}$, outputs a vector $\boldsymbol{e} \in R^{2k}$ sampled from a distribution which is $4(nk)^2\epsilon_s$-close to $D_{\Lambda^\perp_{\mathsf{Coeffs}(u)}\left(\mathsf{Rot}([\boldsymbol{a}^\top|\boldsymbol{b}^\top]\right),\sigma}$, i.e., $[\boldsymbol{a}^\top|\boldsymbol{b}^\top]\cdot\boldsymbol{e} = u$, and $\mathsf{Coeffs}(\boldsymbol{e})$ is distributed according to $D_{\Lambda^\perp_{\mathsf{Coeffs}(u)}\left(\mathsf{Rot}([\boldsymbol{a}^\top|\boldsymbol{b}^\top]\right),\sigma}$.*
- $\mathsf{SampleRight}(\boldsymbol{a},\mathbf{R},u,y,\boldsymbol{g}_b,\mathbf{T}_{\boldsymbol{g}_b},\sigma) \to \boldsymbol{e}$ *where $\boldsymbol{b} = \boldsymbol{a}\mathbf{R} + y\cdot\boldsymbol{g}_b$* ([1]): *A randomized algorithm that, on input the ring vectors $\boldsymbol{a},\boldsymbol{g}_b \in R^k$ such that $\mathsf{Rot}(\boldsymbol{a}^\top), \mathsf{Rot}(\boldsymbol{g}_b^\top) \in \mathbb{Z}^{n\times nk}$ are full-rank, elements $y \in R^*, u \in R$, a matrix $\mathbf{R} \in R^{k\times k}$, a matrix $\mathbf{T}_{\boldsymbol{g}_b} \in R^{k\times k}$ such that $\mathsf{Rot}(\mathbf{T}_{\boldsymbol{g}_b})$ is a basis for the lattice $\Lambda^\perp((\mathsf{Rot}(\boldsymbol{g}_b))$, and a Gaussian parameter $\sigma > s_1(\mathbf{R})\cdot\|\mathsf{Rot}(\mathbf{T}_{\boldsymbol{g}_b})\|_{\mathsf{GS}} \cdot \sqrt{\log(2n(1+1/\epsilon_s))/\pi}$, outputs a vector $\boldsymbol{e} \in R^{2k}$ sampled from a distribution which is $4(nk)^2\epsilon_s$-close to $D_{\Lambda^\perp_{\mathsf{Coeffs}(u)}\left(\mathsf{Rot}([\boldsymbol{a}^\top|\boldsymbol{b}^\top]\right),\sigma}$, i.e., $[\boldsymbol{a}^\top|\boldsymbol{b}^\top]\cdot\boldsymbol{e} = u$, and $\mathsf{Coeffs}(\boldsymbol{e})$ is distributed according to $D_{\Lambda^\perp_{\mathsf{Coeffs}(u)}\left(\mathsf{Rot}([\boldsymbol{a}^\top|\boldsymbol{b}^\top]\right),\sigma}$.*
- ([29]) *Let $k \geq \lceil \log_b(q) \rceil$. There is a publicly known matrix $\mathbf{T}_{\boldsymbol{g}_b}$ such that $\mathsf{Rot}(\mathbf{T}_{\boldsymbol{g}_b})$ is a basis for the lattice $\Lambda^\perp(\mathsf{Rot}(\boldsymbol{g}_b^\top))$ and $\|\mathsf{Rot}(\mathbf{T}_{\boldsymbol{g}_b})\|_{\mathsf{GS}} \leq \sqrt{b^2+1}$. Furthermore, there exists a deterministic polynomial time algorithm $\boldsymbol{g}_b^{-1}$ which takes input $\boldsymbol{u} \in R_q^k$, and outputs $\mathbf{R} = \boldsymbol{g}_b^{-1}(\boldsymbol{u}^\top)$ such that $\mathbf{R} \in [-b,b]_{R^{k\times k}}$, $\boldsymbol{g}_b^\top \cdot \mathbf{R} = \boldsymbol{u}^\top$, and $s_1(\mathbf{R}) \leq nkb$. Similarly, there exists a randomized polynomial time algorithm $\widehat{\boldsymbol{g}_b}^{-1}$ which takes input $\boldsymbol{u} \in R_q^k$, and*

outputs $\mathbf{R} \leftarrow \widehat{\boldsymbol{g}_b}^{-1}(\boldsymbol{u}^\top)$ *such that* $\boldsymbol{g}_b^\top \cdot \mathbf{R} = \boldsymbol{u}^\top$. *Each coefficient in any entry of* $\mathbf{R}$ *follows a sub-Gaussian centered* 0 *with parameter* $O(1)$*, implying* $s_1(\mathbf{R}) \leq \tilde{O}(b\sqrt{nk})$ *with an overwhelming probability.*

Remark 2.12 *Throughout this paper, we make an appointment that* $\boldsymbol{g}_b^{-1}$*(or* $\widehat{\boldsymbol{g}_b}^{-1}$*) maps an integer vector* $\boldsymbol{u} \in \mathbb{Z}_q^k$ *to a integer matrix* $\mathbf{R} \in [-b, b]_{\mathbb{Z}^{k\times k}} \subset \mathbb{Z}^{k\times k}$.

The following lemma shows a simple upper bound of the norm of $\boldsymbol{g}^{-1}(\cdot)$'s output. Due to space limit, we defer the proof in full version of this paper.

Lemma 2.13 *For integers* k, q, b *satisfying the definition of* $\boldsymbol{g}_b$*, on input a vector* $\boldsymbol{c} \in \mathbb{Z}_q^k$*, the algorithm* $\boldsymbol{g}_b^{-1}$ *described in lemma 2.11 outputs the matrix* $\boldsymbol{g}_b^{-1}(\boldsymbol{c}) \in [-b, b]_{R^{k\times k}} \subset R^{k\times k}$ *such that* $\|\boldsymbol{g}_b^{-1}(\boldsymbol{c}^\top)\| \leq bk$.

Homomorphic Computation. In this work, we use the concept of GSW homomorphic encoding [3,21]. We defer the concepts to full version of this paper.

3 New Homomorphic Equality Test and Tighter Analysis

In this section, we present our first main technique – a new homomorphic equality testing method. As discussed in the introduction, our goal can be described as follows: given $\mathsf{Encode}(x^\alpha)$ and $\beta \in \mathbb{Z}$, compute $\mathsf{Encode}(\tau)$ for $\tau = 1$ if $\alpha = \beta$ or otherwise $\tau = 0$. Below we present our method and then an optimization that achieves tighter parameters. Finally, we describe a connection with packing/unpacking GSW encodings using our new technique.

3.1 Homomorphic Equality Testing

As we mentioned in the preliminary, this work focuses on the cyclotomic rings of 2's power, which have simpler mathematical structures. Let $R = \mathbb{Z}[x]/\Phi_m(x)$ be the m-th cyclotomic ring where $m = 2^k$, modulus q be co-prime to m, and $R_q = R/qR$. For this setting, we have $\Phi_m(x) = x^n + 1$ where $n = \varphi(m) = m/2$.

As we discussed in the introduction, we can use the function $f(v) := \sum_{i=0}^{m-1} v^i$ to design an equality tester. Before we formally present the method, we first recall the following important notion that will be used in the design and analysis of our IBE scheme.

Particularly, the lattice IBE framework [1, 4, 25, 39] requires to design two *deterministic* ways to compute the homomorphic encodings. The required property can be formulated in the following notion of δ-expanding evaluation. The parameter δ measures the quality of the evaluation, playing a key factor in the noise analysis of the IBE scheme. Therefore, an important goal in this series of work is to minimize δ from the design and/or analysis.

Definition 3.1 (δ-expanding evaluation [4, 39]) *Two deterministic algorithm* $(\mathsf{PubEval}, \mathsf{TrapEval})$ *are* δ*-expanding with respect to function* $f : \mathcal{X}^t \to \mathcal{Y}$*, if they are efficient and have following properties:*

– $\mathsf{PubEval}(\{\boldsymbol{b}_i \in R_q^k\}_{i\in[t]}, f)$: *on input a function* f *and vectors of encodings* $\{\boldsymbol{b}_i\}_{i\in[t]}$*, this algorithm outputs a ring vector* $\boldsymbol{b}_f \in R_q^k$*;*
– $\mathsf{TrapEval}(\boldsymbol{a} \in R_q^k, \{\mathbf{R}_i \in R^{k\times k}\}_{i\in[t]}, (z_i)_{i\in[t]}, f)$: *the trapdoor evaluation algorithm outputs a matrix* $\mathbf{R}_f \in R^{k\times k}$ *such that for any* $\boldsymbol{z}^\top = (z_1, ..., z_t) \in \mathcal{X}^t$*,* $\boldsymbol{a} \in R_q^k$*, and trapdoor information* $\{\mathbf{R}_i \in R^{k\times k}\}_{i\in[t]}$*:*

$$\mathsf{PubEval}\left(\{\boldsymbol{a}^\top \cdot \mathbf{R}_i + z_i \cdot \boldsymbol{g}_b^\top\}_{i\in[t]}, f\right) = \boldsymbol{a}^\top \cdot \mathbf{R}_f + f(z) \cdot \boldsymbol{g}_b^\top.$$

Furthermore, we have $\|\mathbf{R}_f\| \leq \delta \cdot \max_{i\in[t]} \|\mathbf{R}_i\|$.

This definition can be extended to a family of functions $\mathcal{F}$*, where we require the algorithms to be* δ*-expanding with respect to all functions* $f \in \mathcal{F}$*.*

In the following section, we present our design and analysis for the above equality test function in the term of δ-expanding homomorphic evaluation.

3.2 Our Construction

We first define the family of equality test functions as follow.

Definition 3.2 (Equality Test Function) *Define function* $\mathsf{Equal}_\beta(\cdot)$ *parameterized by* $\beta \in [m]$ *as follows: on input* $x^\alpha \in R$*, the function outputs* 1 *if* $\alpha \equiv \beta \mod m$ *and* 0 *otherwise.*

We next present the algorithms and then analyze the expansion factor.

Construction 3.3 *We present algorithms* $(\mathsf{PubEval}, \mathsf{TrapEval})$ *for* Equal_β *for any* $\beta \in [m]$ *as follows.*

$\mathsf{PubEval}(\{\boldsymbol{b}_\alpha\}, \mathsf{Equal}_\beta)$:
1. *Compute the encoding of* $x^{\alpha-\beta}$ *by* $\boldsymbol{b}' := \boldsymbol{b}_\alpha x^{-\beta}$.
2. *Compute* $\boldsymbol{c}_{m-1}$ *recursively as follows:*

$$\boldsymbol{c}_j = \begin{cases} \boldsymbol{g}_b & j = 0 \\ \boldsymbol{g}_b^{-1}(\boldsymbol{c}_{j-1}^\top)^\top \cdot \boldsymbol{b}' + \boldsymbol{g}_b & j \geq 1 \end{cases} \tag{1}$$

3. *Output* $\boldsymbol{g}_b^{-1}(m^{-1}\boldsymbol{g}_b^\top)^\top \cdot \boldsymbol{c}_{m-1}$.

$\mathsf{TrapEval}(\boldsymbol{a}, \{\mathbf{R}_\alpha\}, (x^\alpha), \mathsf{Equal}_\beta)$:
1. *Compute* $\mathbf{R}' := \mathbf{R}_\alpha \cdot x^{-\beta}$.
2. *Let* $\boldsymbol{c}_j$*'s be vectors as defined as in the* $\mathsf{PubEval}$ *Equation (1) with* $\boldsymbol{b}_\alpha = \boldsymbol{a} \cdot \mathbf{R}_\alpha + x^\alpha \cdot \boldsymbol{g}_b$*. Then compute* $\mathbf{R}_{m-1}$ *recursively as follows:*

$$\mathbf{R}_j := \begin{cases} \mathbf{0} & j = 0 \\ \mathbf{R}' \cdot \boldsymbol{g}_b^{-1}(\boldsymbol{c}_{j-1}^\top) + x^{\alpha-\beta} \cdot \mathbf{R}_{j-1} & j \geq 1 \ , \end{cases} \tag{2}$$

3. *Output* $\mathbf{R}_{m-1} \cdot \boldsymbol{g}_b^{-1}(m^{-1}\boldsymbol{g}_b^\top)$.

In the following theorem, we summarize the quality of the above algorithms. Due to space limit, we put the proof in full version of this paper.

Theorem 3.4 *The algorithms* $(\mathsf{PubEval}, \mathsf{TrapEval})$ *in Construction 3.3 are* $mn(kb)^2$*-expanding with respect to the function family* $\{\mathsf{Equal}_\beta(\cdot)\}_{\beta\in[m]}$.

3.3 An Optimization with Tighter Analysis

In this section, we present an optimization of the above homomorphic evaluation processes that achieves a tighter δ-expansion factor.

We notice that in the IBE settings, we need deterministic evaluation algorithms, so a randomized $\widehat{\boldsymbol{g}_b}^{-1}(\cdot)$ cannot be applied to optimize parameters as the case of FHE evaluation, e.g., [3, 29]. To tackle this challenge, we consider using a randomized $\widehat{\boldsymbol{g}_b}^{-1}$ with a public seed, e.g., a PRF key K. In this way, we can make the $\widehat{\boldsymbol{g}_b}^{-1}$ "deterministic," as everyone can derive the randomness to compute $\widehat{\boldsymbol{g}_b}^{-1}$ from the public key K. Here we notice that we do not use PRF for security, but a way to generate randomness for $\widehat{\boldsymbol{g}_b}^{-1}$. Thus it does not affect the overall security by publishing the seed of PRF in public.

To formalize the idea above, we define a slight variant of δ-homomorphic evaluation in the common random string (CRS) model,[9] where algorithms $(\mathsf{PubEval}, \mathsf{TrapEval})$ have access to a CRS selected randomly in the beginning.

Definition 3.5 (CRS δ-expanding Evaluation) *Algorithms* $(\mathsf{PubEval}, \mathsf{TrapEval})$ *are in the common random string (CRS) model if the algorithms have access to* crs *selected randomly in the beginning. Moreover, they are δ-expanding in the CRS model if with an overwhelming probability (i.e., $1-\mathsf{negl}(\lambda)$) over the choice of* crs*, the algorithms satisfies the requirement of δ-expanding in Definition 3.1.*

Then we can instantiate evaluation algorithms with a tighter δ expanding factor in the CRS model as below. Here we present a sketch.

Construction 3.3 in the CRS Model. Replacing the deterministic $\boldsymbol{g}_b^{-1}$ in Construction 3.3 by a randomized $\widehat{\boldsymbol{g}_b}^{-1}$ under a public PRF key K, we can easily derive $(\mathsf{PubEval}, \mathsf{TrapEval})$ in the CRS model, achieving a better δ parameter. We summarize this optimization in the following theorem.

Theorem 3.6 *There exist* $(\mathsf{PubEval}, \mathsf{TrapEval})$ *that are $\tilde{O}(mb^2k\sqrt{nk})$-expanding in the CRS model for the function family $\{\mathsf{Equal}_\beta(\cdot)\}_{\beta\in[m]}$.*

We defer the details about the construction and analysis to full version of this paper.

An alternative approach and comparison. We notice that the homomorphic equality test can be done if the input is given in the bit representation. Particularly, consider $\mathsf{Equal}'_\beta(\alpha)$ where $\alpha \in [m]$ is given in the form $(\alpha_1, \ldots, \alpha_{\lceil \log m\rceil -1}) \in \{0,1\}^{\lceil \log m\rceil -1}$, then we can express $\mathsf{Equal}'_\beta(\alpha) := \prod_{i=0}^{\lceil \log m\rceil -1}((1-\alpha_i)(1-\beta_i)+\alpha_i\cdot\beta_i)$, where β_i is the i-th bit of $\beta \in [m]$. We can use the method of [3,12,15] for the homomorphic computation, and improve the expanding factor in the CRS model as the above. Particularly we have:

[9] We can define the common reference string model, where crs is selected according to some sampling algorithm. In this work, the common random string model suffices.

Theorem 3.7 *There exist algorithms* $(\mathsf{PubEval}, \mathsf{TrapEval})$ *that are* $O(nkb \log m)$*-expanding in plain model, and are* $\tilde{O}(b\sqrt{nk} \log m)$*-expanding in the CRS model with respect to the function family* $\{\mathsf{Equal}'_\beta(\cdot)\}_{\beta \in [m]}$.

Compared with Construction 3.3, the bit-wise homomorphic evaluation method has a better expanding factor, but would require more input ciphertexts. This would affect our later IBE constructions – our IBE instantiation with Equal_β would require a smaller RLWE $1/\alpha$ (i.e., $1/n^{7.5+O(1)}$) yet smaller mpk, (i.e., $\omega(1)$ basic vectors), and the instantiation with Equal'_β would require a larger RLWE $1/\alpha$ (i.e., $1/n^{4.5+O(1)}$), yet larger mpk, (i.e., $\omega(\log \lambda)$ basic vectors). To our current knowledge, the asymptotic hardness of RLWE does not differ significantly for the two $1/\alpha$'s [2], so the instantiation of IBE with Equal_β as Construction 3.3 has better overall efficiency, asymptotically.

3.4 Application to Packing/Unpacking Homomorphic Encodings

Our equality test technique can be further used to pack/unpack GSW-type [3,21] homomorphic encodings. We defer the details in full version of this paper.

Particularly, we can compress $\log m$ bit-encodings into one encoding of a ring element without losing information. This technique can be generically used to improve FHE [3, 21] for boolean computation, ABE [12] for circuits, and the theoretical state-of-the-arts IBE [39]. As a result , the mpk size in the IBE [39] can be shrunk by a factor of $\log m$ in the ring setting from our technique.

Our technique for the applications demonstrates another non-trivial advantage of RLWE over the plain LWE, which might be of independent interests.

4 New Partition Function and Homomorphic Evaluation

In this section, we describe our second main technique – an explicit design of the partition function required by our IBE scheme and homomorphic evaluation algorithms with a small expansion factor. Our design uses the algebraic structure of cyclotomic rings in a critical way, avoiding the route of Barrington's Theorem as the prior work [39]. As a result, our explicit partition function yields significantly better concrete parameters in the overall IBE scheme.

To describe the partition function, we first recall an insight from the work [4], stating that the IBE design with the trapdoor vanishing technique indeed only needs (weak) pairwise independent hash functions plus the random isolation technique of Valiant and Vazirani [35], which can generically replace the prior notions "admissible hash functions" or "abort-resistant hash functions." We state the following lemma from [4] to summarize this insight.

Lemma 4.1 ([4]) *Let* $Q \subset \{0,1\}^n$ *be an arbitrary subset,* A*,*B *be integers such that* $B \leq A, |Q| \leq \delta B$ *for some* $\delta \in (0,1)$*, and let* $\mathcal{H} : \{0,1\}^n \to \mathcal{Y}$ *be an almost pairwise independent hash function family which has the following properties:*

1 $\forall\ x \in \{0,1\}^n$, $\Pr_{h \in \mathcal{H}}[h(x) = 0] = \frac{1}{A}$.

2 For any distinct $x_1 \neq x_2 \in \{0,1\}^n$, $\Pr_{h\in\mathcal{H}}[h(x_1)=0|h(x_2)=0] < \frac{1}{B}$.

Then for any element $x \notin Q$, *we have*

$$\Pr_{h\in\mathcal{H}}\left[h(x)=0 \wedge \left(h(x')\neq 0 \ \forall x' \in Q\right)\right] \in (\frac{1-\delta}{A}, \frac{1}{A}).$$

Thus, our goal in this section is to (1) design such a hash function family, and (2) design $\mathsf{PubEval}$ and $\mathsf{TrapEval}$ algorithms with a small average-case expanding factor for the hash family. These would suffice for our IBE scheme.

4.1 Our New Hash Function Family

In this section, we first describe a simplified version to illustrate the core idea, and then show how to transform this simplified version to our final design.

Design Idea. Our design uses an error correcting code $\mathsf{ECC} : \mathcal{D} \to \mathbb{Z}_p^{\mathsf{L}}$ with relative distance Υ as follows. We define a basic hash function $h : \mathcal{D} \to \mathbb{Z}_p$:

$$h_{\alpha,\beta}(z) = (\mathsf{ECC}(z)[\alpha] - \beta),$$

where $\alpha \in [\mathsf{L}+1]$ selects the position of $\mathsf{ECC}(z)$ and $\beta \in \mathbb{Z}_p$ represents a shift. Here we use $\{1,\ldots,\mathsf{L}\}$ to index the position of the error correcting code, and assume $\mathsf{ECC}(z)[0] = 0$ for any $z \in \mathcal{D}$. This indexing will be convenient for describing our further constructions.

A hash family is naturally defined as $\mathcal{H} = \{h_{\alpha,\beta} : \alpha \in [\mathsf{L}+1]\setminus\{0\}, \beta \in \mathbb{Z}_p\}$. It is easy to show that (1) $\Pr_{\alpha,\beta}[h_{\alpha,\beta}(z)=0] = 1/p$ for any $z \in \mathcal{D}$ and (2) for any distinct $z_1 \neq z_2 \in \mathcal{D}$, we have $\Pr_{\alpha,\beta}\left[h_{\alpha,\beta}(z_1)=0|h_{\alpha,\beta}(z_2)=0\right] \leq 1-\Upsilon$. Intuitively, for $z_1 \neq z_2$, there is Υ fraction of the positions in their error correcting codes that give different values, meaning with this probability over the choice of α, $\mathsf{ECC}(z_1)[\alpha] \neq \mathsf{ECC}(z_2)[\alpha]$. This would imply $h_{\alpha,\beta}(z_1) \neq h_{\alpha,\beta}(z_2)$, which can be used to derive the probability bound we want.

The basic hash family as is does not yet fulfill what we need for the IBE analysis, as usually the parameter Q (corresponding to the number of adversary's key queries) is larger than the parameter p we can set. To tackle this issue, we use the technique of parallel repetition in the following way. Let $t \in \mathbb{Z}$ be parameter, and $\boldsymbol{\alpha} \in ([\mathsf{L}+1]\setminus\{0\})^t, \boldsymbol{\beta} \in \mathbb{Z}_p^t$ be parameters. We define $h^{\|,t}_{\boldsymbol{\alpha},\boldsymbol{\beta}} : \mathcal{D} \to \mathbb{Z}_p^t$ as

$$h^{\|,t}_{\boldsymbol{\alpha},\boldsymbol{\beta}}(z) = \left(h_{\boldsymbol{\alpha}_1,\boldsymbol{\beta}_1}(z),\ldots,h_{\boldsymbol{\alpha}_t,\boldsymbol{\beta}_t}(z)\right).$$

We can then show that (1) $\Pr_{\boldsymbol{\alpha},\boldsymbol{\beta}}\left[h^{\|,t}_{\boldsymbol{\alpha},\boldsymbol{\beta}}(z)=0\right] = 1/p^t$ and (2) for any distinct $z_1 \neq z_2 \in \mathcal{D}$, we have $\Pr_{\boldsymbol{\alpha},\boldsymbol{\beta}}\left[h^{\|,t}_{\boldsymbol{\alpha},\boldsymbol{\beta}}(z_1)=0|h^{\|,t}_{\boldsymbol{\alpha},\boldsymbol{\beta}}(z_2)=0\right] \leq (1-\Upsilon)^t$.

Thus, by choosing an appropriate parameter t, the family $\mathcal{H}^t = \{h^{\|,t}_{\boldsymbol{\alpha},\boldsymbol{\beta}} : \boldsymbol{\alpha} \in ([\mathsf{L}+1]\setminus\{0\})^t, \boldsymbol{\beta} \in \mathbb{Z}_p^t\}$ and Lemma 4.1 can be used to analyze our IBE security.

Remark 4.2 *As we discussed in the introduction, using error correcting codes to design a partition function has been explored previously in the context of IBE and VRF. e.g., [8,24,39]. Our new insight is to integrate the ECC into the cyclotomic rings so that it can be easily computed homomorphically. More details follow.*

Our Final Construction – Hash in the Ring. However, to design a Ring-based IBE, using the above hash family (as is) still faces two major challenges: (1) the family $\mathcal{H}^t$ with output domain $\mathbb{Z}_p^t$ is not naturally compatible with the ring, and thus not convenient for our ring-based IBE design. (2) The second challenge is quite subtle – the IBE analysis [1,39] requires to compute (homomorphically) $\mathcal{H}^{t'}$ for a flexible $t' \in [t]$, yet in an oblivious way in t', i.e., the evaluation only depends on t but does not know t'. The purpose is to derive a more fine-grained security analysis for the IBE scheme. Therefore, the hash family must at least capture $\cup_{t' \in [t]} \mathcal{H}^{t'}$, and support this type of oblivious evaluation.

To tackle these issues, we propose a modified ring-based hash family $\mathcal{H}^{R,t}$ that captures all $\mathcal{H}^{t'}$ for $t' \leq t$ and matches the output domain with the ring R of the RLWE. At a high level, $\mathcal{H}^{R,t}$ embeds $\mathcal{H}^{t'}$ with output $\mathbb{Z}_p^{t'}$ for all $t' \in [t]$ into some subset of the ring R, which is naturally compatible with our Ring IBE design. Next we present our final design, starting with some important notations.

Important Notations. Let R be the m-th cyclotomic ring and $n = m/2$; p, t be integers such that $tp \leq n$; $\mathsf{ECC} : \mathcal{D} \to \mathbb{Z}_p^{\mathsf{L}}$ with relative distance Υ be an error correcting code whose codeword is indexed by $\{1, \ldots, \mathsf{L}\}$ and $\mathsf{ECC}(z)[0] = 0$ for every $z \in \mathcal{D}$. Then, we present our design of the hash function as follow.

Definition 4.3 *For any $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \in [\mathsf{L}+1]^t \times \mathbb{Z}_p^t$, we define hash function $H^{R,t}_{\boldsymbol{\alpha},\boldsymbol{\beta}} : \mathcal{D} \to R$ as $H^{R,t}_{\boldsymbol{\alpha},\boldsymbol{\beta}}(z) := \sum_{i \in [t]} \left(x^{ip + \mathsf{ECC}(z)[\boldsymbol{\alpha}_i]} - x^{ip+\boldsymbol{\beta}_i} \right)$.*

According to the property $\mathsf{ECC}(z)[0] = 0$, in the above hash we extend the range of $\boldsymbol{\alpha}$ to $[\mathsf{L}+1]^t$ without affecting the result. Under this design, we define the following classes of hash functions:

Definition 4.4 *For any $t' \in [t]$, define the class $\mathcal{H}^{R,t,t'}$ as follows.*

$$\mathcal{H}^{R,t,t'} = \left\{ H^{R,t}_{\boldsymbol{\alpha},\boldsymbol{\beta}} : \boldsymbol{\alpha}' \in ([\mathsf{L}+1] \setminus \{0\})^{t'}, \boldsymbol{\beta}' \in \mathbb{Z}_p^{t'}, \boldsymbol{\alpha}^\top = (\boldsymbol{\alpha}'^\top, \mathbf{0}^\top), \boldsymbol{\beta}^\top = (\boldsymbol{\beta}'^\top, \mathbf{0}^\top) \right\},$$

where $\mathbf{0}^\top = (0, 0, \ldots, 0) \in \mathbb{Z}_p^{t-t'}$, i.e., padding 0's to match the dimension t. Furthermore, define $\mathcal{H}^{R,t} = \cup_{t' \in [t]} \mathcal{H}^{R,t,t'}$.

Intuitively, for a fixed t', if the index $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is chosen randomly from the set $([\mathsf{L}+1] \setminus \{0\})^t \times \mathbb{Z}_p^t$, then the function $H^{R,t}$ behaves like $h^{\|,t'}$ as we elaborate next. Observe that we can view $H^{R,t}_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ as a hash that embeds the vector $h^{\|,t'}_{\boldsymbol{\alpha},\boldsymbol{\beta}} \in \mathbb{Z}_p^{t'}$ into the ring R. From our setting that $\mathsf{ECC}(z)[0] = 0$, the padded $\mathbf{0}$'s will result in cancelled terms in $H^{R,t}$, i.e., $x^{ip + \mathsf{ECC}(z)[0]} - x^{ip} = 0$ for every $i \in [t'+1, t]$. Moreover, we notice that different coordinates in the output vector of $h^{\|,t'}$ will not interfere – the i-th coordinate of the vector, namely $h_{\boldsymbol{\alpha}_i,\boldsymbol{\beta}_i}(z)$, corresponds to the ring element $x^{ip+\mathsf{ECC}(z)[\boldsymbol{\alpha}_i]} - x^{ip+\boldsymbol{\beta}_i}$. As both $\mathsf{ECC}(z)[\boldsymbol{\alpha}_i]$ and $\boldsymbol{\beta}_i$ take values between 0 and $p-1$, our design guarantees that $(x^{ip+\mathsf{ECC}(z)[\boldsymbol{\alpha}_i]} - x^{ip+\boldsymbol{\beta}_i})$ would not interfere with $(x^{jp+\mathsf{ECC}(z)[\boldsymbol{\alpha}_i]} - x^{jp+\boldsymbol{\beta}_i})$ for $i \neq j$. Formally we prove the following lemma.

Lemma 4.5 *For any code* $\mathsf{ECC} : \mathcal{D} \to \mathbb{Z}_p^{\mathsf{L}}$ *with relative distance* Υ*, ring* R *with dimension* n *such that* $tp \leq n$*, Then for any* $t' \leq t$*, the hash function family* $\mathcal{H}^{R,t,t'}$ *as in Definition 4.4 has following properties:*

1 *For any element* $z_1 \in \mathcal{D}$, $\underset{H \in \mathcal{H}^{R,t,t'}}{\mathsf{Pr}}\left[H(z_1) = 0\right] = (1/p)^{t'}$.

2 *For any distinct elements* $z_1 \neq z_2 \in \mathcal{D}$*, we have*

$$\underset{H \in \mathcal{H}^{R,t,t'}}{\mathsf{Pr}}\left[H(z_1) = 0 | H(z_2) = 0\right] < (1 - \Upsilon)^{t'}.$$

We defer the proof of this lemma in full version of this paper.

Two Further Important Properties. It is important to point out two further important properties that will be used in our IBE analysis.

- (**Obliviousness**) The computation of the hash $H^{R,t}_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ is oblivious to the choice of $(\boldsymbol{\alpha}, \boldsymbol{\beta})$. That is to say, for any $t' \leq t$ and any choice of $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \in ([\mathsf{L}+1] \setminus \{0\})^t \times \mathbb{Z}_p^t$
  the way to compute $H^{R,t}_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ remains the same. This is extremely important for our IBE design and proof of security.
- (**Invertibility**) We notice that if $H^{R,t}(z) \neq 0$, then it is also invertible in the ring R_q for any prime $q \equiv 3 \mod 8$ and $q \geq 2t$. This is because $\|H^{R,t}(z)\|_2 \leq \sqrt{2t} \leq \sqrt{q}$. By Lemma 2.7, any element with norm less than $\sqrt{q}$ is invertible in R_q for this type of prime q.

4.2 Homomorphic Evaluation of the Partitioning Function

To homomorphically evaluate the hash function, we first identify the high level goal: given input encodings $\{\mathsf{Encode}(x^{\boldsymbol{\alpha}_i})\}_{i\in[t]}$, $\mathsf{Encode}\left(\sum_{i\in[t]} x^{ip+\boldsymbol{\beta}_i}\right)$ and a hash input $z \in \mathcal{D}$ in the clear, our task is to output an encoding $\mathsf{Encode}\left(H^{R,t}_{\boldsymbol{\alpha},\boldsymbol{\beta}}(z)\right)$.

To achieve this, we first observe that we can re-write the hash function as

$$H^{R,t}_{\boldsymbol{\alpha},\boldsymbol{\beta}}(z) = -\sum_{i\in[t]} x^{ip+\boldsymbol{\beta}_i} + \sum_{i\in[t]} \sum_{j\in[\mathsf{L}+1]} \left(j \overset{?}{=} \boldsymbol{\alpha}_i\right) x^{ip+\mathsf{ECC}(z)[j]},$$

where $\left(j \overset{?}{=} \boldsymbol{\alpha}_i\right)$ outputs 1 if the equality holds and otherwise 0. Recall that we index the codeword by $[1, \mathsf{L}]$ and we set $\mathsf{ECC}(z)[0] = 0$ for any $z \in \mathcal{D}$.

As the input z and iterators i, j are in the clear, the only non-trivial homomorphic computation is the equality test $\left(j \overset{?}{=} \boldsymbol{\alpha}_i\right)$. The reader at this point might already observe that this is what we achieved in the prior Construction 3.3, if we further have $\mathsf{L} + 1 \leq m$ (as our equality test function naturally only supports comparison of parameters in $[m]$). However, our application would require longer codewords, i.e., $\mathsf{L} = m^\eta > m + 1$ for some $\eta > 1$, so this direct approach would not work. To solve this issue, we consider input encodings $\{\mathsf{Encode}(\boldsymbol{\alpha}_{i,i'})\}_{i\in[t],i'\in[\eta]}$ where $(\boldsymbol{\alpha}_{i,0}, \ldots, \boldsymbol{\alpha}_{i,(\eta-1)})$ is considered as the m-ary

representation of $\boldsymbol{\alpha}_i \in [\mathsf{L}+1] \setminus \{0\}$. To test whether $j \stackrel{?}{=} \boldsymbol{\alpha}_i$ for $j \in [\mathsf{L}+1] \setminus \{0\}$, we can first compute the m-ary representation of j as $(j_0, \ldots, j_{\eta-1})$ and then check whether $j_{i'} = \boldsymbol{\alpha}_{i,i'}$ for every $i' \in [\eta]$.

Using this insight, we present the procedure formally. To work under the syntax of $(\mathsf{PubEval}, \mathsf{TrapEval})$, we define the hash function in the following form where the computation is in the clear:

Definition 4.6 (Hash Function for Homomorphic Evaluation) *Let R be some cyclotomic ring with degree n being a power of 2, q be an integer, $R_q = R/qR$ and ECC be an error correcting code mapping $\mathcal{D} \to \mathbb{Z}_p^{\mathsf{L}}$, satisfying the constraint $tp \leq n$ and further $\mathsf{L}+1 \leq m^\eta$. Suppose the function $\mathsf{Equal}_\beta(x^\alpha)$ parametered by β outputs $1 \in \mathbb{Z}_q$ if the input x^α satisfying $\alpha = \beta$ and $0 \in \mathbb{Z}_q$ otherwise. Define function $F_z(\{\boldsymbol{\alpha}_{i,i'}\}_{i\in[t],i'\in[\eta]}, \tilde{\beta})$ parameterized by $z \in \mathcal{D}$ as: on input $\{\boldsymbol{\alpha}_{i,i'}\}_{i\in[t],i'\in[\eta]} \in [m]^{t\times\eta}$, $\tilde{\beta} \in R_q$, the function computes as follows.*

- *For each $j \in [\mathsf{L}+1]$, denote j's m-ary representation as $(j[0], \ldots, j[\eta-1])$.*
- *For each $i \in [t], j \in [\mathsf{L}+1]$, compute $b_{i,j} = \prod_{i'\in[\eta]} \mathsf{Equal}_{j[i']}(x^{\boldsymbol{\alpha}_{i,i'}})$.*
- *Output $-\tilde{\beta} + \sum_{i\in[t],j\in[\mathsf{L}+1]} b_{i,j} \cdot x^{ip+\mathsf{ECC}(z)[j]}$.*

Under the above notation, we present the homomorphic evaluation procedures.

Construction 4.7 *Given $(\mathsf{PubEval}, \mathsf{TrapEval})$ for $\{\mathsf{Equal}_\beta(\cdot)\}_{\beta\in[m]}$ (either in the plain or CRS model; ref. Sections 3.2 and 3.3) as subroutine, we construct $(\mathsf{PubEval}, \mathsf{TrapEval})$ for $\{F_z\}_{z\in\mathcal{D}}$ (in the plain or CRS model, respectively) as:*

$\mathsf{PubEval}\left(\left\{\{\boldsymbol{b}_{\boldsymbol{\alpha}_{i,i'}}\}_{i\in[t],i'\in[\eta]}, \boldsymbol{b}_{\tilde{\beta}}\right\}, F_z\right)$**:**

1 *For $i \in [t], j \in [\mathsf{L}+1], i' \in [\eta]$, (homomorphically) compute*

$$\boldsymbol{b}_{i,j,i'} = \begin{cases} \mathsf{PubEval}(\boldsymbol{b}_{\boldsymbol{\alpha}_{i,0}}, \mathsf{Equal}_{j[0]}) & i' = 0, \\ \mathsf{PubEval}(\boldsymbol{b}_{\boldsymbol{\alpha}_{i,i'}}, \mathsf{Equal}_{j[i']}) \cdot \boldsymbol{g}_b^{-1}\left(\boldsymbol{b}_{i,j,(i'-1)}\right) & i' \geq 1. \end{cases}$$

Then, let $\boldsymbol{b}_{i,j} := \boldsymbol{b}_{i,j,(\eta-1)}$

2 *Output $\boldsymbol{b}_H := -\boldsymbol{b}_{\tilde{\beta}} + \sum_{i\in[t],j\in[\mathsf{L}+1]} \boldsymbol{b}_{i,j} \cdot x^{ip+\mathsf{ECC}(z)[j]}$*

$\mathsf{TrapEval}\left(\boldsymbol{a}, \left\{\{\mathbf{R}_{\boldsymbol{\alpha}_{ii'}}\}_{i\in[t],i'\in[\eta]} \subset R_q^{k\times k}, \mathbf{R}_{\tilde{\beta}} \in R_q^{k\times k}\right\}, (x^{\boldsymbol{\alpha}}, \tilde{\beta}), F_z\right)$**:**

1 *For $i \in [t], j \in [\mathsf{L}+1], i' \in [\eta]$, (homomorphically) compute*

$$\mathbf{R}'_{i,j,i'} := \mathsf{TrapEval}\left(\boldsymbol{a}, \{\mathbf{R}_{\boldsymbol{\alpha}_{i,i'}}\}, (x^{\boldsymbol{\alpha}_{i,i'}}), \mathsf{Equal}_{j[i']}\right).$$

2 *For $i' \in [\eta]$, let $\boldsymbol{b}_{i,j,i'}$ be the vector evaluated in $\mathsf{PubEval}$ algorithm with $\boldsymbol{b}_{\boldsymbol{\alpha}_{i,i'}} = \boldsymbol{a} \cdot \mathbf{R}_{\boldsymbol{\alpha}_{ii'}} + x^{\boldsymbol{\alpha}_{i,i'}} \cdot \boldsymbol{g}_b$, and recursively compute*

$$\mathbf{R}_{i,j,i'} = \begin{cases} \mathbf{R}'_{i,j,0} & i' = 0, \\ \mathbf{R}'_{i,j,i'} \cdot \boldsymbol{g}_b^{-1}\left(\boldsymbol{b}_{i,j,i'-1}\right) + \mathsf{Equal}_{j[i']}(x^{\boldsymbol{\alpha}_{i,i'}}) \cdot \mathbf{R}_{i,j,i'-1} & i' \geq 1. \end{cases}$$

Then let $\mathbf{R}_{i,j} := \mathbf{R}_{i,j,(\eta-1)}$.

3 *Output* $\mathbf{R}_H := -\mathbf{R}_{\tilde{\beta}} + \sum_{i\in[t],j\in[\mathsf{L}+1]} \mathbf{R}_{i,j}$.

We can easily calculate the expansion factor for the above $(\mathsf{PubEval}, \mathsf{TrapEval})$ for the family $\{F_z\}_{z\in\mathcal{D}}$, assuming we have $(\mathsf{PubEval}, \mathsf{TrapEval})$ that is δ-expanding for the family $\{\mathsf{Equal}_j\}_{j\in[m]}$, either in the plain or CRS model. We present the detailed analysis in full version of this paper.

Moreover, we notice that for the case $\eta = 1$, i.e., $\mathsf{L}+1 \leq m$, we do not need to do the m-ary decomposition, and thus can obtain a better expanding factor by avoiding several layers of homomorphic multiplications. By combining Theorems 3.4 and 3.6 with the above construction, we can obtain the following corollary, showing the existence of the algorithms $(\mathsf{PubEval}, \mathsf{TrapEval})$ respect to the function family $\{F_z\}_{z\in\mathcal{D}}$ in both plain and CRS models.

Corollary 4.8 *Consider parameters $tp \leq n$ and $\mathsf{L}+1 \leq m^\eta$ and others as stated in Definition 4.6. Then there exist an algorithm pair $(\mathsf{PubEval}, \mathsf{TrapEval})$ with following two properties:*

1. *If $\eta = 1$, the algorithms are $(\mathsf{L}+1)tmn(kb)^2$-expanding in the plain model, and $\tilde{O}(t\mathsf{L}mkb^2\sqrt{nk})$-expanding in the CRS model for the family $\{F_z\}_{z\in\mathcal{D}}$.*
2. *If $\eta > 1$, the algorithms are $(\mathsf{L}+1)tmn^2(kb)^3\eta$-expanding in the plain model, and $\tilde{O}(t\mathsf{L}mnk^2b^3\eta)$-expanding in the CRS model for the family $\{F_z\}_{z\in\mathcal{D}}$.*

Alternatively, if we use the bit-wise equality test computation (i.e., $\mathsf{Equal}'_\beta()$) as the underlying building block, then by Theorems 3.7 with the above construction, we can obtain the following corollary.

Corollary 4.9 *Consider parameters $tp \leq n$ and others as stated in Definition 4.6. There exist an algorithm pair $(\mathsf{PubEval}, \mathsf{TrapEval})$ that are $(\mathsf{L}+1)tnkb\log m$-expanding in the plain model, and $\tilde{O}(t\mathsf{L}b\sqrt{nk}\log m)$-expanding in the CRS model for the family $\{F_z\}_{z\in\mathcal{D}}$.*

5 IBE Design and Analysis

Now we present the design and improvement of analysis of IBE.

5.1 Construction

Our IBE construction uses the building block – algorithms $(\mathsf{PubEval}, \mathsf{TrapEval})$ for the function class $\{F_z\}_{z\in\mathcal{D}}$ as Construction 4.7. We note that the function class requires an error correcting code $\mathsf{ECC} : \{0,1\}^\ell \to \mathbb{Z}_p^{\mathsf{L}}$ where $\mathsf{L}+1 \leq m^\eta$. Next we present the construction.

Construction 5.1 *For identity space $\mathsf{ID} = \{0,1\}^\ell$ and message space $\mathsf{M} = \{0,1\}^n$, we define IBE scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ as follows:*

$\mathsf{Setup}(1^\lambda)$: *On input security parameter 1^λ, the Setup algorithm does:*

1. *Sample $(\boldsymbol{a}, \mathbf{T}_{\boldsymbol{a}}) \leftarrow \mathsf{TrapGen}(n, k, \rho, q)$, where $\boldsymbol{a} \in R_q^k$.*

2. *Choose* $\eta t + 1$ *random ring vectors, i.e.,* $\boldsymbol{b}_{i,j} \xleftarrow{\$} R_q^k$ *for* $i \in [t], j \in [\eta]$, $\boldsymbol{b}_\beta \xleftarrow{\$} R_q^k$, *and a random ring element* $u \xleftarrow{\$} R_q$.
3. *Sample a* PRF *key* K *as the CRS for the homomorphic evaluation.*
4. *Output the master keys as:* $\mathsf{mpk} = (\boldsymbol{a}, (\boldsymbol{b}_{i,j})_{i\in[t],j\in[\eta]}, \boldsymbol{b}_\beta, u, K), \mathsf{msk} = \mathbf{T}_{\boldsymbol{a}}$.

$\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$*: On input the master keys* $\mathsf{mpk}, \mathsf{msk}$ *and an identity* $\mathsf{id} \in \mathsf{ID}$*, the* KeyGen *algorithm does the following:*

1. *Define* F_{id} *as the function as in Definition 4.6 with index* id.
2. *Compute* $\boldsymbol{b}_{\mathsf{id}} = \mathsf{PubEval}\big(\{\boldsymbol{b}_{i,j}\}_{i\in[t],j\in[\eta]}, \boldsymbol{b}_\beta, F_{\mathsf{id}}\big)$.
3. *Sample* $\boldsymbol{r} \in R_q^{2k}$ *by* $\mathsf{SampleLeft}(\boldsymbol{a}, \boldsymbol{b}_{\mathsf{id}}, \boldsymbol{T}_{\boldsymbol{a}}, u, \sigma_1)$*, satisfying* $\boldsymbol{r}^\top \cdot \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b}_{\mathsf{id}} \end{bmatrix} = u$.
4. *Output* $\mathsf{sk}_{\mathsf{id}} = \boldsymbol{r}$ *as a secret key of* id.

$\mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, \boldsymbol{m})$*: On input* mpk, id *and message* $\boldsymbol{m} \in \mathsf{M}$*, the algorithm does:*

1. *Set* $\mu = m_0 + m_1 x + \cdots + m_{n-1} x^{n-1} \in R_q$.
2. *Compute* $\boldsymbol{b}_{\mathsf{id}} = \mathsf{PubEval}(\{\boldsymbol{b}_{i,j}\}_{i\in[t],j\in[\eta]}, \boldsymbol{b}_\beta, F_{\mathsf{id}})$.
3. *Sample* $s \xleftarrow{\$} R_q$*, and sample* $\boldsymbol{e}_1, \boldsymbol{e}_2 \leftarrow (D^{\mathsf{Coeffs}}_{\mathbb{Z}^n,\sigma_2})^k$ *and* $e_3 \leftarrow D^{\mathsf{Coeffs}}_{\mathbb{Z}^n,\sigma_{\mathsf{RLWE}}}$.
4. *Compute* $c_0 = u \cdot s + e_3 + \lceil \frac{q}{2} \rceil \cdot \mu$*, and* $\boldsymbol{c}_1 = \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b}_{id} \end{bmatrix} \cdot s + \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \end{bmatrix}$.
5. *Output the ciphertext* $\mathsf{ct} = (c_0, \boldsymbol{c}_1) \in R_q \times R_q^{2k}$.

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}}, \mathsf{ct})$ *On input the master public key* mpk*, the secret key* $\mathsf{sk}_{\mathsf{id}} = \boldsymbol{r}$ *and ciphertext* $\mathsf{ct} = (c_0, \boldsymbol{c}_1)$*, the decryption algorithm does the following:*

1. *Output* $\boldsymbol{m}' = \lfloor \frac{2}{q} \cdot \mathsf{Coeffs}(c_0 - \boldsymbol{r}^\top \cdot \boldsymbol{c}_1) \rceil \bmod 2$*, where the rounding function* $\lfloor \cdot \rceil$ *is applied coefficient-wise.*

Correctness. Correctness of our IBE scheme is captured by the following Theorem. We defer the proof of it in full version of our paper.

Theorem 5.2 *For any positive number* ω*, and ring modulus* $q \geq 5(\sigma_{\mathsf{RLWE}} \cdot \omega + \sigma_1\sigma_2\sqrt{2nk} \cdot \omega)$*, the* IBE *scheme* Π *presented in construction 5.1 is correct except with probability* $2^{-2nk+2} + 4e^{-\pi\omega^2}$.

5.2 Security

In this section, we analyze security of our IBE construction. Below we first present a theorem for a reduction from RLWE to IBE with concrete parameters.

We first define and recall several notations. Let $\mathsf{ECC} : \{0,1\}^\ell \to \mathbb{Z}_p^{\mathsf{L}}$ with relative distance Υ be the underlying error correcting code of the function family $\{F_{\mathsf{id}}\}_{\mathsf{id}\in\mathsf{ID}}$. We denote $c = 1/(1-\Upsilon)$. For our instantiations, we have the relations $\mathsf{L} + 1 \leq m^\eta$ and $p > c > p/w$ for some small $w \in \mathbb{R}$, which can be set between $[2, \lambda]$ depending on the selection of the code. We denote ϵ_s as a small positive real regarding the smoothing parameter involved in SampleLeft and SampleRight algorithms. Asymptotically, we would set $\epsilon_s = \mathsf{negl}(\lambda)$, and concretely $\epsilon_s =$

$2^{-3\lambda}$, ensuring that the parameter ε defined below satisfies $\varepsilon = \mathsf{negl}(\lambda)$ or $\varepsilon \leq \frac{1}{2^{2\lambda}}$. Intuitively, this means the statistical distance incurred in the sampling algorithms (in the scheme and proof of security) would be negligible or bounded by $\varepsilon \leq \frac{1}{2^{2\lambda}}$. Then we have the following theorem.

Theorem 5.3 *Given any (T, α, β)-adversary $\mathcal{A}$ making Q' key queries against $\mathbf{Expt}_{\mathcal{A}}^{\mathsf{IBE}}(1^\lambda)$, there exists a (T', α', β')-adversary $\mathcal{B}$ against $\mathbf{Expt}_{\mathcal{B}}^{\mathsf{RLWE}}(1^n, k+1, q, \psi)$, such that $T' \leq T + \min\{O\left(\frac{Q' p w^{t'-1}}{(2\beta-1)^2}\right), (\mathsf{L}p)^{t'}\}$, $\alpha' \geq \frac{(5-2\beta)\alpha}{36Q' p w^{t'-1}} - \frac{1}{2}\varepsilon$, and $|\beta' - \frac{1}{2}| \geq \frac{1}{2}\left(\frac{11-6\beta}{8}\beta - (\frac{5-2\beta}{36Q' p w^{t'-1}\varepsilon} - 1)^{-1}\right) - \frac{1}{4}$, where $\varepsilon = \frac{k}{2^{\frac{nk}{4}}} + (Q'(nk)^2 + 1)8\epsilon_s$, $t' = \lceil \log_c(3Q') \rceil$.*

As discussed in the introduction, our analysis improves the running time of the artificial abort technique of Waters [37]. We present the proof below.

Proof. Let $\mathcal{A}$ be a (T, α, β)-adversary who makes Q' key queries against the IBE game of $\mathbf{Expt}_{\mathcal{A}}^{\mathsf{IBE}}(1^\lambda)$, and our goal is to construct a RLWE adversary $\mathcal{B}$ that satisfies the parameters as the theorem statement. Before presenting the concrete construction of $\mathcal{B}$, we first define several hybrids, from which the design idea of $\mathcal{B}$ naturally reveals.

Hybrid 0: In this hybrid, $\mathcal{A}$ plays the original security experiment $\mathbf{Expt}_{\mathcal{A}}^{\mathsf{IBE}}(1^\lambda)$.

Hybrid 1: In this hybrid, $\mathcal{A}$ plays a slightly modified security experiment $\mathbf{Expt}_{\mathcal{A}}^{\mathsf{IBE}}(1^\lambda)'$ where the challenger has an additional ability to send a $\perp$ message to $\mathcal{A}$ at any step, and then $\mathcal{A}$ would immediately abort upon receiving this message. The particular modified experiment is defined as follows:

- The setup phase is identical to $\mathbf{Expt}_{\mathcal{A}}^{\mathsf{IBE}}(1^\lambda)$ except that the challenger chooses a random partitioning function $H \xleftarrow{\$} \mathcal{H}^{R,t,t'}$ as Definition 4.4, where $t' = \lceil \log_c(3Q') \rceil$. Particularly, the challenger would sample random vectors $\boldsymbol{\alpha}' \in [\mathsf{L}+1]^{t'}, \boldsymbol{\beta}' \in \mathbb{Z}_p^{t'}$, denotes $\boldsymbol{\alpha} = (\boldsymbol{\alpha}', \mathbf{0}) \in [\mathsf{L}+1]^t, \boldsymbol{\beta} = (\boldsymbol{\beta}', \mathbf{0}) \in \mathbb{Z}_p^t$, and finally sets and keeps the hash function:

$$H(\mathsf{id}) = F_{\mathsf{id}}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = H_{\boldsymbol{\alpha},\boldsymbol{\beta}}^{R,t}(\mathsf{id}) = \sum_{i \in [t]} (x^{ip + \mathsf{ECC}(\mathsf{id})[\boldsymbol{\alpha}_i]} - x^{ip+\boldsymbol{\beta}_i}).$$

- The challenger responds to identity queries and issues the challenge ciphertext exactly as in $\mathbf{Expt}_{\mathcal{A}}^{\mathsf{IBE}}(1^\lambda)$. Let $\mathsf{id}_1, \cdots, \mathsf{id}_{Q'}$ be the identities where the attacker queries and let id^* be the challenge identity, which is not in $\{\mathsf{id}_1, \cdots, \mathsf{id}_{Q'}\}$.
- In the final phase, adversary $\mathcal{A}$ might output a bit b' as its guess or might have aborted at some prior step. If the adversary does not abort, then the challenger does the *abort check* and *artificial abort* as follow:
  1. **Abort check:** the challenger checks if:

$$H(\mathsf{id}) \neq 0 \text{ for all } \mathsf{id} \in Q', \text{ and } H(\mathsf{id}^*) = 0.$$

  If the condition does not hold, challenger sends $\perp$ to $\mathcal{A}$, and $\mathcal{A}$ will abort the game upon receiving $\perp$.

2. **Artificial abort:** the challenger samples a bit $\Gamma \in \{0,1\}$ such that $\mathsf{Pr}[\Gamma = 1] = 1 - \tilde{\gamma}(\mathsf{id}^*, \mathsf{id}_1, \cdots, \mathsf{id}_{Q'})$ where $\tilde{\gamma}(\cdot)$ is defined as follows:
   - Define γ to be the probability as follow:
   $$\gamma = \Pr_{H \in \mathcal{H}^{R,t,t'}} \left[H(\mathsf{id}_i) \neq 0 \text{ for all } i \leq Q', \text{ and } H(\mathsf{id}^*) = 0 \right]. \quad (3)$$
   - If $O\left(\frac{\log(2\beta-1)\cdot\log(\gamma^*)}{(2\beta-1)^2\gamma^*}\right) < (\mathsf{L}p)^{t'}$, then the challenger samples $O\left(\frac{\log(2\beta-1)\cdot\log(\gamma^*)}{(2\beta-1)^2\gamma^*}\right)$ pairs of $(\boldsymbol{\alpha}', \boldsymbol{\beta}')$ and computes the hash values of $H^{R,t}_{\boldsymbol{\alpha},\boldsymbol{\beta}}(*)$ for the identities $(\mathsf{id}_1, \cdots, \mathsf{id}_{Q'}, \mathsf{id}^*)$ to compute an estimate γ' of γ, where $\gamma^* = \frac{2}{9Q'pw^{t'-1}}$. Otherwise, challenger computes the *exact value* of γ by enumerating all choices of $\boldsymbol{\alpha}, \boldsymbol{\beta}$'s of the hash function for $(\mathsf{id}_1, \cdots, \mathsf{id}_{Q'}, \mathsf{id}^*)$. Notice that there are $(\mathsf{L}p)^{t'}$ choices of $(\boldsymbol{\alpha}, \boldsymbol{\beta})$. Set $\gamma' = \gamma$.
   - If $\gamma^* \leq \gamma'$, challenger sets $\tilde{\gamma}(\mathsf{id}^*, \mathsf{id}_1, \cdots, \mathsf{id}_{Q'}) = \gamma^*/\gamma'$, otherwise sets $\tilde{\gamma}(\mathsf{id}^*, \mathsf{id}_1, \cdots, \mathsf{id}_{Q'}) = 1$.

   If $\Gamma = 1$ the challenger sends $\perp$ to $\mathcal{A}$, and then $\mathcal{A}$ aborts the game. In this case we say that the challenger aborted the game due to an artificial abort.

Hybrid 2: In this hybrid, $\mathcal{A}$ plays $\mathbf{Expt}^{\mathsf{IBE}}_{\mathcal{A}}(1^\lambda)'$ the same as Hybrid 1 except for changing the way of generating the public vectors $\{\boldsymbol{b}_{i,j}\}_{i\in[t],j\in[\eta]}, \boldsymbol{b}_\beta$. Here, the challenger chooses $\boldsymbol{\alpha}_i \in [\mathsf{L}+1]$ for $i \in [t]$ as Hybrid 1, and additionally $\mathbf{R}_{i,j}, \mathbf{R}_\beta \leftarrow [-\rho, \rho]^{k\times k}_R$ for $i \in [t], j \in [\eta]$. For each $i \in [t]$, the challenger further decomposes $\boldsymbol{\alpha}_i \in [\mathsf{L}+1]$ into the m-ary representation $(\boldsymbol{\alpha}_{i,0}, \ldots, \boldsymbol{\alpha}_{i,(\eta-1)})$. Then define the public matrices as follows:

$$\boldsymbol{b}^\top_{i,j} = \boldsymbol{a}^\top \cdot \mathbf{R}_{i,j} + x^{\boldsymbol{\alpha}_{i,j}} \cdot \boldsymbol{g}^\top_b \quad \text{and} \quad \boldsymbol{b}^\top_\beta = \boldsymbol{a}^\top \cdot \mathbf{R}_\beta + \sum_{i\in[t]} x^{ip+\boldsymbol{\beta}_i} \cdot \boldsymbol{g}^\top_b.$$

Hybrid 3: In this hybrid, $\mathcal{A}$ plays $\mathbf{Expt}^{\mathsf{IBE}}_{\mathcal{A}}(1^\lambda)'$ the same as Hybrid 2 except that we change the way to generate the public vector $\boldsymbol{a}$ and to respond the secret key queries. Formally, the challenger samples $\boldsymbol{a} \xleftarrow{\$} R^k_q$ uniformly at random instead of running TrapGen algorithm. On the other hand, to respond a secret key query for id, the challenger first computes

$$\mathbf{R}_{\mathsf{id}} = \mathsf{TrapEval}(\boldsymbol{a}, \{\mathbf{R}_{i,j}\}_{i\in[t],j\in[\eta]}, \mathbf{R}_\beta, (\boldsymbol{\alpha}, \boldsymbol{\beta}), F_{\mathsf{id}}).$$

By the homomorphic property, we know that $\boldsymbol{b}^\top_{\mathsf{id}} = \mathsf{PubEval}\big(\{\boldsymbol{b}_{i,j}\}_{i\in[t],j\in[\eta]}, \boldsymbol{b}_\beta, F_{\mathsf{id}}\big) = \boldsymbol{a}^\top \cdot \mathbf{R}_{\mathsf{id}} + F_{\mathsf{id}}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \cdot \boldsymbol{g}^\top_b$. Then the challenger runs

$$\boldsymbol{r} \leftarrow \mathsf{SampleRight}(\boldsymbol{a}, \mathbf{R}_{\mathsf{id}}, u, F_{\mathsf{id}}(\boldsymbol{\alpha}, \boldsymbol{\beta}), \boldsymbol{g}_b, \mathbf{T}_{\boldsymbol{g}_b}, \sigma)$$

satisfying $[\boldsymbol{a}^\top | \boldsymbol{b}^\top_{\mathsf{id}}] \cdot \boldsymbol{r} = u \bmod q$. Finally, the challenger outputs the secret key $\mathsf{sk}_{\mathsf{id}} = \boldsymbol{r} \in R^{2k}_q$.

Hybrid 4: In this hybrid, $\mathcal{A}$ plays $\mathbf{Expt}^{\mathsf{IBE}}_{\mathcal{A}}(1^\lambda)'$ the same as Hybrid 3 except for the way that challenge ciphertext $(c^*_0, \boldsymbol{c}^*_1)$ is generated. The challenger first

chooses $s \xleftarrow{\$} R_q$, $\boldsymbol{x} \leftarrow (D^{\mathsf{Coeffs}}_{\mathbb{Z}^n,\sigma_{\mathsf{RLWE}}})^k$ and sets $\boldsymbol{v} = \boldsymbol{a} \cdot s + \boldsymbol{x} \in R_q^k$. Then, the challenger samples $e_3 \leftarrow D^{\mathsf{Coeffs}}_{\mathbb{Z}^n,\sigma_{\mathsf{RLWE}}}$, and sets the challenge ciphertext as

$$c_0^* = u \cdot s + e_3 + \lceil \frac{q}{2} \rceil \mu \quad \text{and} \quad \boldsymbol{c}_1^* = \mathsf{ReRand}\big([\mathbf{I}_k | \mathbf{R}_{\mathsf{id}^*}]^\top, \boldsymbol{v}, \sigma_{\mathsf{RLWE}}, \sigma_3\big),$$

where $\mathbf{I}_k$ is the identity matrix in $R^{k\times k}$ and $\sigma_3 = \frac{\sigma_2}{2\sigma_{\mathsf{RLWE}}}$. The syntax of the re-randomization algorithm is defined in full version of this paper.

Hybrid 5: In this hybrid, $\mathcal{A}$ plays $\mathbf{Expt}^{\mathsf{IBE}}_{\mathcal{A}}(1^\lambda)'$ the same as Hybrid 4 except for the way that the challenge ciphertext is generated. Here, the challenger first chooses random c_0 from R_q and random $\boldsymbol{v}'$ from R_q^k, and samples $\boldsymbol{x} \leftarrow (D^{\mathsf{Coeffs}}_{\mathbb{Z}^n,\sigma_{\mathsf{RLWE}}})^k$. Then challenger sets the challenge ciphertext as

$$c_0^* = c_0 + \lceil \frac{q}{2} \rceil \mu \quad \text{and} \quad \boldsymbol{c}_1^* = \mathsf{ReRand}\big([\mathbf{I}_k | \mathbf{R}_{\mathsf{id}^*}]^\top, \boldsymbol{v}, \sigma_{\mathsf{RLWE}}, \sigma_3\big),$$

where $\boldsymbol{v} = \boldsymbol{v}' + \boldsymbol{x}$, and σ_3 is defined as in Hybrid 4. As c_0 is uniformly random and independent of $\boldsymbol{c}_1^*$, it serves as a one-time pad that perfectly hides μ. Thus the advantage of the adversary in this hybrid is exactly 0.

Next we are going to analyze the adversary's advantage in each hybrid. Similar as the previous analysis, we denote (T_i, α_i, β_i) as $\mathcal{A}$'s running time, non-abort probability, and successfully conditional guessing probability in Hybrid i for $0 \le i \le 5$. We note that $(T_0, \alpha_0, \beta_0) = (T, \alpha, \beta)$ by the condition of the theorem, and $\beta_5 = 1/2$ as the message is hidden by an one-time pad in Hybrid 5. Particularly, we derive the following lemmas. Due to space limit, we defer the proofs of Lemma 5.4, 5.5, 5.6 in full version of our paper.

Lemma 5.4 *$T_1 = T_0 + \min\left\{O\left(\frac{\log(2\beta-1)\cdot\log(\gamma^*)}{(2\beta-1)^2\gamma^*}\right), (\mathsf{L}p)^{t'}\right\}$, $\alpha_1 \ge \alpha\gamma^*\cdot(1-\frac{2\beta-1}{4})$, and $\beta_1 \ge \big(1-\frac{3}{8}(2\beta-1)\big)\cdot\beta$, where $\gamma^* = \frac{2}{9Q'pw^{t'-1}}$.*

Lemma 5.5 *$T_1 = T_4$, $\alpha_4 \ge \alpha_1 - \varepsilon$ and $\beta_4 \ge \beta_1 - \varepsilon/(\alpha_1 - \varepsilon)$.*

Lemma 5.6 *There exists a (T', α', β')-adversary $\mathcal{B}$ against $\mathbf{Expt}^{\mathsf{RLWE}}_{\mathcal{B}}(1^n, k+1, q, \psi)$ such that $T' \le T_4 + O\left(\frac{\log(2\beta-1)\cdot\log(\gamma^*)}{(2\beta-1)^2\gamma^*}\right)$, $\alpha' \ge \alpha_4/2$ and $\beta' \ge \beta_4/2 + 1/4$, where $\gamma^* = \frac{2}{9Q'pw^{t'-1}}$.*

Combining Lemma 5.4, 5.5 and Lemma 5.6, it's easy to verify that $T' \le T + \min\{O(\frac{\log(2\beta-1)\cdot\log(\gamma^*)}{(2\beta-1)^2\gamma^*}), (\mathsf{L}p)^{t'}\}$, $\alpha' \ge \frac{1}{2}(\alpha\gamma^*\cdot(1-\frac{2\beta-1}{4})-\varepsilon) \ge \frac{(5-2\beta)\alpha}{36Q'pw^{t'-1}} - \frac{1}{2}\varepsilon$ and $\beta' \ge \frac{1}{2}((1-\frac{6\beta-3}{8})\beta - (\gamma^*\alpha(1-\frac{2\beta-1}{4})/\varepsilon - 1)^{-1}) + \frac{1}{4} \ge \frac{1}{2}\left(\frac{11-6\beta}{8}\beta - (\frac{5-2\beta}{36Q'pw^{t'-1}\varepsilon} - 1)^{-1}\right) + \frac{1}{4}$, and thus we have that $|\beta' - 1/2| \ge \frac{1}{2}\left(\frac{11-6\beta}{8}\beta - (\frac{5-2\beta}{36Q'pw^{t'-1}\varepsilon} - 1)^{-1}\right) - \frac{1}{4}$. This completes the proof. □

5.3 Asymptotic and Concrete Parameters

We also describe how to set both asymptotic and concrete parameters for our IBE scheme in the full version of this paper. Due to space limit, we summarize the results as follows:

Corollary 5.7 (Asymptotic Parameterization) *Assume* RLWE *is hard for parameters* $n = \Theta(\lambda), 1/\alpha := \sigma_{\mathsf{RLWE}}/q = 1/\mathsf{poly}(\lambda)$. *Then Construction 5.1 is an adaptively secure* IBE. *The reductions cost* (T', ϵ') *satisfies* $T' = T + \min\{O\left(p^t \log(p^t) \cdot \log(1/\epsilon)/\epsilon\right), (\mathsf{L}p)^t\}$, $\epsilon' \geq O\big(\epsilon/\lambda^{\frac{1}{\kappa}} Q\big)$, *where* T *and* ϵ *are the running time and advantage of an* IBE *adversary, who makes* Q *key queries.*

Corollary 5.8 (Concrete Parameterization) *Assume the* RLWE *is* $\max\{\lambda + \left\lceil \lceil \log_c(3Q) \rceil \cdot \frac{1}{\kappa} \log n \right\rceil + 10, \left\lceil \lceil \log_c(3Q) \rceil \cdot \frac{\kappa+4}{\kappa} \log n + \log(\frac{1}{\epsilon}) \right\rceil + 10\}$*-bit hard for parameters* $n, q, \sigma_{\mathsf{RLWE}}$, *where* $c = 1/(1-\Upsilon) = \sqrt[\kappa]{n}/(\kappa+3)$, *and* ϵ *is the advantage of an* IBE *adversary. Then Construction 5.1 is an adaptively secure* IBE, *and can achieve* λ*-bit security.*

Acknowledgement. We would like to thank the anonymous reviewers of TCC 2021 for their insightful advices. Feng-Hao Liu and Zhedong Wang are supported by an NSF Award CNS-1657040 and an NSF Career Award CNS-1942400. Part of this work was done while Zhedong Wang was a postdoc at Florida Atlantic University. Parhat Abla and Han Wang are supported by the National Natural Science Foundation of China under Grant Number NSFC61772516 and the National Key R&D Program of China under Grant Number 2020YFA0712303. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsors.

References

1. S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In Gilbert [22], pages 553–572.
2. M. R. Albrecht, R. Player, and S. Scott. On the concrete hardness of learning with errors. Journal of Mathematical Cryptology. Volume 9, Issue 3, Pages 169–203, 2015. `https://bitbucket.org/malb/lwe-estimator/src/master/`.
3. J. Alperin-Sheriff and C. Peikert. Faster bootstrapping with polynomial error. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 297–314. Springer, Heidelberg, Aug. 2014.
4. D. Apon, X. Fan, and F.-H. Liu. Vector encoding over lattices and its applications. Cryptology ePrint Archive, Report 2017/455, 2017. `http://eprint.iacr.org/2017/455`.
5. D. A. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in nc1. *Journal of Computer and System Sciences*, 38(1):150–164, 1989.
6. P. W. Beame, S. A. Cook, and H. J. Hoover. Log depth circuits for division and related problems. *SIAM Journal on Computing*, 15(4):994–1003, 1986.
7. 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, Heidelberg, Apr. 2009.

8. N. Bitansky. Verifiable random functions from non-interactive witness-indistinguishable proofs. In Y. Kalai and L. Reyzin, editors, *TCC 2017, Part II*, volume 10678 of *LNCS*, pages 567–594. Springer, Heidelberg, Nov. 2017.
9. D. Boneh and X. Boyen. Efficient selective-ID secure identity based encryption without random oracles. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238. Springer, Heidelberg, May 2004.
10. D. Boneh and X. Boyen. Secure identity based encryption without random oracles. In M. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 443–459. Springer, Heidelberg, Aug. 2004.
11. D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. In J. Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, Heidelberg, Aug. 2001.
12. D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, and D. Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 533–556. Springer, Heidelberg, May 2014.
13. X. Boyen and Q. Li. Towards tightly secure lattice short signature and id-based encryption. In Cheon and Takagi [17], pages 404–434.
14. Z. Brakerski, A. Lombardi, G. Segev, and V. Vaikuntanathan. Anonymous IBE, leakage resilience and circular security from new assumptions. In Nielsen and Rijmen [31], pages 535–564.
15. Z. Brakerski and V. Vaikuntanathan. Lattice-based FHE as secure as PKE. In M. Naor, editor, *ITCS 2014*, pages 1–12. ACM, Jan. 2014.
16. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In Gilbert [22], pages 523–552.
17. J. H. Cheon and T. Takagi, editors. *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*. Springer, Heidelberg, Dec. 2016.
18. N. Döttling and S. Garg. From selective IBE to full IBE and selective HIBE. In Y. Kalai and L. Reyzin, editors, *TCC 2017, Part I*, volume 10677 of *LNCS*, pages 372–408. Springer, Heidelberg, Nov. 2017.
19. C. Gentry. Practical identity-based encryption without random oracles. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 445–464. Springer, Heidelberg, May / June 2006.
20. 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.
21. C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 75–92. Springer, Heidelberg, Aug. 2013.
22. H. Gilbert, editor. *EUROCRYPT 2010*, volume 6110 of *LNCS*. Springer, Heidelberg, May / June 2010.
23. S. Gorbunov and D. Vinayagamurthy. Riding on asymmetry: Efficient ABE for branching programs. In T. Iwata and J. H. Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 550–574. Springer, Heidelberg, Nov. / Dec. 2015.
24. S. Katsumata. On the untapped potential of encoding predicates by arithmetic circuits and their applications. In T. Takagi and T. Peyrin, editors, *ASI-*

ACRYPT 2017, Part III, volume 10626 of *LNCS*, pages 95–125. Springer, Heidelberg, Dec. 2017.
25. S. Katsumata and S. Yamada. Partitioning via non-linear polynomial functions: More compact IBEs from ideal lattices and bilinear maps. In Cheon and Takagi [17], pages 682–712.
26. 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, Heidelberg, May 2020.
27. V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In Gilbert [22], pages 1–23.
28. V. Lyubashevsky, C. Peikert, and O. Regev. A toolkit for ring-LWE cryptography. In T. Johansson and P. Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 35–54. Springer, Heidelberg, May 2013.
29. 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, Heidelberg, Apr. 2012.
30. D. Micciancio and M. Walter. On the bit security of cryptographic primitives. In Nielsen and Rijmen [31], pages 3–28.
31. J. B. Nielsen and V. Rijmen, editors. *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*. Springer, Heidelberg, Apr. / May 2018.
32. 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.
33. A. Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and D. Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 47–53. Springer, Heidelberg, Aug. 1984.
34. D. Stehlé, R. Steinfeld, K. Tanaka, and K. Xagawa. Efficient public key encryption based on ideal lattices. In M. Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 617–635. Springer, Heidelberg, Dec. 2009.
35. L. G. Valiant and V. V. Vazirani. NP is as easy as detecting unique solutions. In *17th ACM STOC*, pages 458–463. ACM Press, May 1985.
36. 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, Heidelberg, Aug. 2009.
37. B. R. Waters. Efficient identity-based encryption without random oracles. In R. Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127. Springer, Heidelberg, May 2005.
38. 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, Heidelberg, May 2016.
39. 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, Heidelberg, Aug. 2017.
40. 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, Heidelberg, Aug. 2016.