Lattice-based Programmable Hash Functions and Applications

Jiang Zhang[1], Yu Chen[1,2], and Zhenfeng Zhang[3]

[1] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China
[2] Key Laboratory of Cryptologic Technology and Information Security of Ministry of Education, School of Cyber Science and Technology, Shandong University, Qingdao 266237, China
[3] Trusted Computing and Information Assurance Laboratory,
Institute of Software, Chinese Academy of Sciences, China
jiangzhang09@gmail.com, yuchen.prc@gmail.com, zfzhang@tca.iscas.ac.cn

Abstract. Driven by the open problem raised by Hofheinz and Kiltz (Journal of Cryptology, 2012), we study the formalization of lattice-based programmable hash function (PHF), and give three types of concrete constructions by using several techniques such as a novel combination of cover-free sets and lattice trapdoors. Under the Inhomogeneous Small Integer Solution (ISIS) assumption, we show that any (non-trivial) lattice-based PHF is a collision-resistant hash function, which gives a direct application of this new primitive.

We further demonstrate the power of lattice-based PHF by giving generic constructions of signature and identity-based encryption (IBE) in the standard model, which not only provide a way to unify several previous lattice-based schemes using the partitioning proof techniques, but also allow us to obtain new short signature schemes and IBE schemes from (ideal) lattices. Specifically, by instantiating the generic constructions with our Type-II and Type-III PHF constructions, we immediately obtain two short signatures and two IBE schemes with asymptotically much shorter keys. A major downside which inherits from our Type-II and Type-III PHF constructions is that we can only prove the security of the new signatures and IBEs in the bounded security model that the number Q of the adversary's queries is required to be known in advance. Another downside is that the computational time of our new signatures and IBEs is a linear function of Q, which is large for typical parameters.

To overcome the above limitations, we also give a refined way of using Type-II and Type-III PHFs to construct lattice-based short signatures with short verification keys in the full security model. In particular, our methods depart from the confined guessing technique of Böhl et al. (Eurocrypt'13) that was used to construct previous standard model short signature schemes with short verification keys by Ducas and Micciancio (Crypto'14) and by Alperin-Sheriff (PKC'15), and allow us to achieve much tighter security from weaker hardness assumptions.

Content

1 Introduction

As a primitive capturing the partitioning proof techniques, programmable hash function introduced by Hofheinz and Kiltz [35] is a powerful tool to construct provably secure cryptographic schemes in the standard model. Informally, a PHF $\mathcal{H} = \{\mathrm{H}_K\}$ is a keyed group hash function over some finite group $\mathbb{G}$, which can work in two (statistically) indistinguishable modes depending on how the key is generated: if the key K is generated in the normal mode, then the hash function behaves normally and maps an input X into a group element $\mathrm{H}_K(X) \in \mathbb{G}$; while if the key K' is generated in the trapdoor mode, then with the help of some trapdoor information td it can additionally output a secret pair (a_X, b_X) such that $\mathrm{H}_{K'}(X) = g^{a_X} h^{b_X}$ holds for some prior fixed group generators $g, h \in \mathbb{G}$. More formally, let $u, v \in \mathbb{Z}$ be some positive integers, $\mathcal{H}$ is said to be (u, v)-programmable if given any inputs $X_1, \ldots, X_u$ and $Y_1, \ldots, Y_v$ satisfying $X_i \neq Y_j$ for any i and j, the probability $\Pr[a_{X_1} = \cdots = a_{X_u} = 0 \wedge a_{Y_1}, \ldots, a_{Y_v} \neq 0] \geq 1/\mathsf{poly}(\kappa)$ for some polynomial $\mathsf{poly}(\kappa)$ in the security parameter κ, where the probability is over the random coins used in generating K' and td. This feature gives a partition of all inputs in terms of whether $a_X = 0$, and becomes very useful in security proofs when the discrete logarithm (DL) is hard in $\mathbb{G}$ [35].

Since its introduction, PHFs have attracted much attention from the research community [58,33,36,28,17], and had been used to construct many cryptographic schemes (such as short signature schemes [34]) in the standard model. However, both the definition and the constructions of traditional PHFs seem specific to hash functions defined over groups where the "DL problem" is hard. This might be the reason why almost all known PHFs were constructed from "DL groups". Actually, it was left as an open problem [36] to find instantiations of PHF from different assumptions, e.g., lattices.

Facing the rapid development of quantum computers, the past decade has witnessed remarkable advancement in lattice-based cryptography. Nevertheless, the silhouette of lattice-based PHFs is still not very clear. At Crypto 2013, Freire et al. [28] extended the notion of PHF to the multilinear maps setting. However, recent study shows that there is a long way to go before obtaining a practical and secure multilinear maps from lattices [29,21,18,20,37]. An intriguing question of great interest is to construct lattice-based PHFs or something similar based on standard hard lattice problems.

Lattice-based Short Signatures. It is well-known that digital signature schemes [41] can be constructed from general assumptions, such as one-way functions. Nevertheless, these generic signature schemes suffer from either large signatures or large verification keys, thus a main open problem is to reduce the signature size as well as the verification key size. The first direct constructions of lattice-based signature schemes were given in [45,31]. Later, many works (e.g., [44,24,7]) significantly improved the efficiency of lattice-based signature schemes in the random oracle model. In comparison, the progress in constructing efficient lattice-based signature schemes in the standard model was relatively slow. At

Eurocrypt 2010, Cash et al. [16] proposed a signature scheme with a linear number of vectors in the signatures. The first standard model short signature scheme with signatures consisting of a single lattice vector was due to Boyen [13], which was later improved by Micciancio and Peikert [47]. However, the verification keys of both schemes in [13,47] consist of a linear number of matrices.

In 2013, Böhl et al. [9] constructed a lattice-based signature scheme with constant verification keys by introducing the confined guessing proof technique. Later, Ducas and Micciancio [26] adapted the confined guessing proof technique to ideal lattices, and proposed a short signature scheme with logarithmic verification keys. Alperin-Sheriff [6] constructed a short signature with constant verification keys based on a stronger hardness assumption by using the idea of homomorphic trapdoor functions [32]. Due to the use of the confined guessing technique, the above three signature schemes [9,26,6] shared two undesired byproducts. First, the security can only be directly proven to be existentially unforgeable against non-adaptive chosen message attacks (EUF-naCMA). Even if an EUF-naCMA secure scheme can be transformed into an EUF-CMA secure one by using known techniques such as chameleon hash functions [42], in the lattice setting [26] this usually introduces an additional tag to each signature and roughly doubles the signature size. Second, a reduction loss about $(Q^2/\epsilon)^c$ for some parameter $c > 1$ seems unavoidable, where Q is the number of signing queries of the forger $\mathcal{F}$, and ϵ is the success probability of $\mathcal{F}$. Therefore, it is desirable to directly construct an EUF-CMA secure scheme that has short signatures, short verification keys, as well as a relatively tight security proof.

Identity-based Encryption from Lattices. Shamir [52] introduced identity-based encryption (IBE) in 1984, but the first realizations were due to Boneh and Franklin from pairings [11] and Cocks from quadratic residues [19]. In the lattice setting, Gentry et al. [31] proposed the first IBE scheme based on the learning with errors (LWE) assumption in the random oracle model. Later, several works [3,16,59,25] were dedicated to the study of lattice-based (hierarchical) IBE schemes also in the random oracle model. There were a few works focusing on designing standard model lattice-based IBE schemes [2,3,16]. Concretely, the scheme in [3] was only proven to be *selective-identity* secure in the standard model. By using standard complexity leverage technique [10], one can generally transform a selective-identity secure IBE scheme into a *fully secure* one. But the resulting scheme has to suffer from a reduction loss proportional to L, where L is the number of distinct identities for the IBE system and is independent from the number Q of the adversary's private key queries in the security proof. Since L is usually super-polynomial and much larger than Q, the above generic transformation is a very unsatisfying approach [30]. In [2,16], the authors showed how to achieve *full security* against adaptive chosen-plaintext and chosen-identity attacks, but both standard model fully secure IBE schemes in [2,16] had large master public keys consisting of a linear number of matrices. In fact, Agrawal, Boneh and Boyen left it as an open problem to find fully secure lattice-based IBE schemes with short master public keys in the standard model [2].

1.1 Our Contributions

Because of the (big) differences in the algebraic structures between lattices and DL groups, the traditional definition of PHFs does not seem to work on lattices. This makes it highly non-trivial to find instantiations of traditional PHFs on lattices. In this paper, we introduce the notion of lattice-based programmable hash function (PHF). Although our lattice-based PHF has gone beyond the realm of traditional PHFs, we prefer to still name it as PHF because it inherits the concept of traditional PHFs and aims at capturing the partitioning proof trick on lattices. By carefully exploiting the algebraic properties of lattices, we give three types of concrete constructions of lattice-based PHFs.

Under the Inhomogeneous Small Integer Solution (ISIS) assumption, we show that any (non-trivial) lattice-based PHF is collision-resistant. This gives a direct application of lattice-based PHFs. We further demonstrate the power of lattice-based PHFs by showing a generic way to construct short signature schemes. We also give a generic IBE scheme from lattice-based PHFs. Moreover, our IBE scheme can be extended to support hierarchical identities, and achieve chosen ciphertext security.

We find that lattice-based PHFs are implicitly used as the backbones in the signature schemes [13,47] and the IBE schemes [2]. Therefore, our results provide a way to unify and clarify those lattice-based cryptographic schemes using the partitioning proof strategy. Furthermore, by instantiating the generic schemes with our Type-II and Type-III PHF constructions, we immediately obtain several new short signature schemes and IBE schemes with shorter keys. A drawback inherited from our concrete Type-II and Type-III PHF constructions of using cover-free sets is that we can only prove the security of those schemes in a bounded security model which requires the number Q of the adversary's queries to be known in advance. Another downside is that the computational time of our new signatures and IBEs is a linear function of Q, and thus is very large for typical choice of parameters.

To remove the above limitations, we also propose two concrete short signature schemes with shorter verification keys and relatively tighter reductions in the full security model by combining the confined guessing technique in [9] with our Type-II and Type-III PHFs. Comparisons between our schemes and previous ones will be given in Section 1.4 and Section 1.5.

1.2 Additions to the Conference Version

This article is a significantly extended and improved version of the conference paper [60]. In addition to providing full proofs for some theorems, we also add the following contributions to this new version:

Extending to ideal lattices. We extend the definition and the concrete constructions of lattice-based PHF given in [60] to general rings so that they are more compatible to ideal lattices.

New lattice-based PHFs. In addition to two concrete PHF constructions in [60], we give a new (namely, Type-III) construction of lattice-based PHF on ideal lattices, which achieves constant keys and thus is $O(\log n)$ times smaller than previous ones. We also present an improved and optimized Type-III PHF construction for some restricted but useful parameters.

Two fully-secure short signatures. In addition to a new short signature instantiated from the generic construction in [60] and our new Type-III PHF, we also propose two signature schemes in the full security model by combining the confined guessing technique in [9] with our Type-II and Type-III PHFs, which seem to be asymptotically the best known lattice-based short signatures with shorter keys in terms of the reduction loss and the hardness parameter in the full security model.

Improved generic IBE construction. We present an improved and simplified generic construction of IBE from lattice-based PHFs, which removes the high min-entropy requirement on the underlying lattice-based PHFs [60], and thus supports more flexible choices of parameters. By instantiating it with our Type-III PHF, we immediately obtain a new IBE scheme with constant master public keys from ideal lattices (in the bounded security model as that in [60]).

1.3 Techniques

We introduce the notion of lattice-based PHFs by carefully exploiting the specific algebraic structure of lattices. Formally, our notion of lattice-based PHFs is defined over some ring $\mathcal{R}$, which can be either the integer ring $\mathbb{Z}$ for general lattices or the polynomial ring $R = \mathbb{Z}[x]/(x^n + 1)$ for ideal lattices. By $\mathcal{R}_q$ we denote the quotient ring $\mathcal{R}/q\mathcal{R}$. As the traditional PHFs, our lattice-based PHF $\mathcal{H} = \{\mathrm{H}_K\}$ can work in two modes. Given a key K generated in either the normal mode or the trapdoor mode, the hash function H_K maps its input $X \in \mathcal{X}$ into a matrix $\mathrm{H}_K(X) \in \mathcal{R}_q^{n\times m}$ for some positive $n, m, q \in \mathbb{Z}$. In the trapdoor mode, there additionally exists a secret trapdoor td allowing to compute matrices $\mathbf{R}_X \in \mathcal{R}_q^{\bar{m}\times m}$ and $\mathbf{S}_X \in \mathcal{R}_q^{n\times n}$ for some integer $\bar{m} \in \mathbb{Z}$, such that $\mathrm{H}_K(X) = \mathbf{A}\mathbf{R}_X + \mathbf{S}_X\mathbf{B} \in \mathcal{R}_q^{n\times m}$ holds with respect to user-specified "generators" $\mathbf{A} \in \mathcal{R}_q^{n\times \bar{m}}$ and $\mathbf{B} \in \mathcal{R}_q^{n\times m}$. For non-triviality, we require that the keys generated in the two modes are statistically indistinguishable (even conditioned on the matrix $\mathbf{A}$ that was used to generate the trapdoor mode key), and that the two "generators" $\mathbf{A} \in \mathcal{R}_q^{n\times \bar{m}}$ and $\mathbf{B} \in \mathcal{R}_q^{n\times m}$ have essential differences for embedding hard lattice problems. More precisely, in our definition $\mathbf{A} \in \mathcal{R}_q^{n\times \bar{m}}$ is required to be uniformly distributed (and thus can be used to embed the ISIS problem over $\mathcal{R}$), while $\mathbf{B} \in \mathcal{R}_q^{n\times m}$ is a trapdoor matrix that allows to efficiently sample short vector $\mathbf{e} \in \mathcal{R}^m$ satisfying $\mathbf{B}\mathbf{e} = \mathbf{v}$ for any vector $\mathbf{v} \in \mathcal{R}_q^n$.

In order to explore the differences between $\mathbf{A} \in \mathcal{R}_q^{n\times \bar{m}}$ and $\mathbf{B} \in \mathcal{R}_q^{n\times m}$ in the security reduction, we require that the largest singular value $s_1(\mathbf{R}_X)$ of $\mathbf{R}_X$ is small, and that $\mathbf{S}_X \in \mathcal{I}_n \cup \{\mathbf{0}\}$ where $\mathcal{I}_n$ is the set of invertible matrices in $\mathcal{R}_q^{n\times n}$.

More concretely, for any positive integer $u, v \in \mathbb{Z}$ and real $\beta \in \mathbb{R}$, a (u, v, β)-PHF $\mathcal{H}$ should satisfy the following two conditions: 1) $s_1(\mathbf{R}_X) \leq \beta$ holds for any input X; and 2) given any inputs $X_1, \ldots, X_u$ and $Y_1, \ldots, Y_v$ satisfying $X_i \neq Y_j$ for any i and j, the probability $\Pr[\mathbf{S}_{X_1} = \cdots = \mathbf{S}_{X_u} = \mathbf{0} \wedge \mathbf{S}_{Y_1}, \ldots, \mathbf{S}_{Y_v} \in \mathcal{I}_n]$ is at least $1/\mathsf{poly}(\kappa)$, where the probability is taken over the random coins in producing td and K'. Besides, if the second condition only holds for some prior fixed $X_1, \ldots, X_u$ (chosen before generating the trapdoor mode key K'), we say that the hash function $\mathcal{H}$ is a weak (u, v, β)-PHF.

Looking ahead, if the trapdoor mode key K' is generated by using $\mathbf{A} \in \mathcal{R}_q^{n \times \bar{m}}$ and trapdoor matrix $\mathbf{B} \in \mathcal{R}_q^{n \times m}$, then for any input X the matrix $\mathbf{A}_X := (\mathbf{A} \| \mathrm{H}_{K'}(X)) = (\mathbf{A} \| \mathbf{A}\mathbf{R}_X + \mathbf{S}_X \mathbf{B}) \in \mathcal{R}_q^{n \times (\bar{m}+m)}$ has a trapdoor $\mathbf{R}_X$ with respect to tag $\mathbf{S}_X$. The programmability comes from the fact that such a trapdoor enables us to sample short vector $\mathbf{e}$ satisfying $\mathbf{A}_X \mathbf{e} = \mathbf{v}$ for any vector $\mathbf{v} \in \mathcal{R}_q^n$ when $\mathbf{S}_X$ is invertible, and loses this ability when $\mathbf{S}_X = \mathbf{0}$. This gives us the possibility to adaptively embed the ISIS problem depending on each particular input X. Since this feature is only useful when the key K' is used together with the "generator" $\mathbf{A} \in \mathcal{R}_q^{n \times \bar{m}}$, we require the keys in both modes to be statistically indistinguishable even conditioned on the information of $\mathbf{A}$.

Type-I PHF Construction. Our Type-I PHF construction is a high-level abstraction of the functions that were (implicitly) used in both signature schemes (e.g, [13,9,47]) and encryption schemes (e.g., [2,47]). Formally, let E be an encoding function from some domain $\mathcal{X}$ to $(\mathcal{R}_q^{n \times n})^\ell$, where ℓ is an integer. Then, for any input $X \in \mathcal{X}$, the Type-I PHF construction $\mathcal{H} = \{\mathrm{H}_K\}$ from $\mathcal{X}$ to $\mathbb{Z}_q^{n \times m}$ is defined as $\mathrm{H}_K(X) = \mathbf{A}_0 + \sum_{i=1}^{\ell} \mathbf{C}_i \mathbf{A}_i$, where $K = (\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_\ell)$ and $\mathrm{E}(X) = (\mathbf{C}_1, \ldots, \mathbf{C}_\ell)$. For appropriate choices of parameters and encoding function E, the literature (implicitly) showed that the Type-I construction satisfies our definition of lattice-based PHFs. Concretely, if one sets $\mathcal{R} = \mathbb{Z}, \mathcal{X} = \{0,1\}^\ell$, and $\mathrm{E}(X) = ((-1)^{X_1} \cdot \mathbf{I}_n, \ldots, (-1)^{X_\ell} \cdot \mathbf{I}_n)$ for any input $X = (X_1, \ldots, X_\ell)$, where $\mathbf{I}_n$ is the $n \times n$ identity matrix. Then, the instantiated PHF is exactly the hash functions that were used to construct the signature scheme in [13] and the IBE scheme in [2]. Since the Type-I PHF construction is independent from the particular choice of $\mathbf{B} \in \mathcal{R}_q^{n \times m}$, it allows us to use any trapdoor matrix $\mathbf{B}$ when generating the trapdoor mode key. However, such a construction has a large key size, i.e., the number of matrices in the key is linear in the input length ℓ.

Type-II PHF Construction. Our Type-II PHF construction has keys only consisting of $O(\log \ell)$ matrices, which substantially reduces the key size by using a novel combination of the cover-free sets and the publicly known trapdoor matrix $\mathbf{B} = \mathbf{G}$ in [47], where $\mathbf{G} = \mathbf{I}_n \otimes \mathbf{g}^T \in \mathcal{R}_q^{n \times nk}$, $k = \lceil \log_2 q \rceil$ and $\mathbf{g} = (1, 2, \ldots, 2^{k-1})^T \in \mathcal{R}_q^k$. Concretely, for any positive $L \in \mathbb{Z}$, by $[L]$ we denote the set $\{0, 1, \ldots, L-1\}$. Recall that if $CF = \{CF_X\}_{X \in [L]}$ is a family of v-cover-free sets over domain $[N]$ for some integers $v, L, N \in \mathbb{Z}$, then for any subset $\mathcal{S} \subseteq [L]$ of size at most v and any $Y \notin \mathcal{S}$, there is at least one element $z^* \in CF_Y \subseteq [N]$ that is not included in the union set $\cup_{X \in \mathcal{S}} CF_X$. The property of cover-free sets naturally gives a partition of $[L]$, and was first used in constructing traditional PHFs in [34]. However, a direct application of the cover-free sets in constructing (lattice-based) PHFs will result in a very large key size (which is even worse than that of the Type-I PHF). Actually, for an input size $L = 2^\ell$, the key of the PHF in [34] should contain an associated element for each element in $[N]$, where ℓ is the input length and $N = O(v^2\ell)$. We solve this problem by using the nice property of $\mathbf{G}$ and the binary representation of the cover-free sets. Formally, let $\mathbf{G}^{-1}(\mathbf{C})$ be the binary decomposition of some matrix $\mathbf{C}$. By the definition of $\mathbf{G}$, we have $\mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{C}) = \mathbf{C}$. Now, we set the key K of the Type-II PHF as $K = (\mathbf{A}, \{\mathbf{A}_i\}_{i \in \{0,\ldots,\mu-1\}})$, where $\mu = \lceil \log_2 N \rceil = O(\log \ell)$ for some prior fixed polynomial $v = \mathsf{poly}(\ell)$ in the input length ℓ. Given an input $X \in [L] = [2^\ell]$, we first map X into the corresponding set $CF_X \in CF$. Then, for each $z \in CF_X \subseteq [N]$, we "recover" an associated matrix $\mathbf{A}_z = \mathsf{Func}(K, z, 0)$ from K and the binary decomposition $(b_0, \ldots, b_{\mu-1})$ of z, where Func is recursively defined as

$$\mathsf{Func}(K, z, i) = \begin{cases} \mathbf{A}_{\mu-1} - b_{\mu-1}\mathbf{G}, \text{ if } i = \mu - 1 \\ (\mathbf{A}_i - b_i\mathbf{G}) \cdot \mathbf{G}^{-1}(\mathsf{Func}(K, z, i+1)), \text{ otherwise} \end{cases}$$

Finally, we output the hash value $\mathrm{H}_K(X) = \mathbf{A} + \sum_{z \in CF_X} \mathbf{A}_z$.

In the trapdoor mode, we randomly choose a "target" element $z^* \in [N]$, and set $\mathbf{A} = \hat{\mathbf{A}}\mathbf{R} - (-1)^c \cdot \mathbf{G}$ and $\mathbf{A}_i = \hat{\mathbf{A}}\mathbf{R}_i + (1 - b_i^*) \cdot \mathbf{G}$ for all $i \in \{0, \ldots, \mu - 1\}$, where $(b_0^*, \ldots, b_{\mu-1}^*)$ is the binary decomposition of z^* and c is the number of 1's in the vector $(b_0^*, \ldots, b_{\mu-1}^*)$. By doing this, we have that $\mathbf{A}_z = \hat{\mathbf{A}}\hat{\mathbf{R}}_z + \hat{\mathbf{S}}_z\mathbf{G}$ holds for some matrices $\hat{\mathbf{R}}_z$ and $\hat{\mathbf{S}}_z = \prod_{i=0}^{\mu-1}(1 - b_i^* - b_i) \cdot \mathbf{I}_n$, where $(b_0, \ldots, b_{\mu-1})$ is the binary decomposition of z. This means that $\hat{\mathbf{S}}_z = \mathbf{0}$ for any $z \neq z^*$, and $\hat{\mathbf{S}}_{z^*} = (-1)^c \cdot \mathbf{I}_n$. By the definition of $\mathrm{H}_K(X) = \mathbf{A} + \sum_{z \in CF_X} \mathbf{A}_z$, we have that $\mathrm{H}_K(X) = \hat{\mathbf{A}}\hat{\mathbf{R}}_X + \hat{\mathbf{S}}_X\mathbf{G}$ holds for some matrices $\hat{\mathbf{R}}_X = \mathbf{R} + \sum_{z \in CF_X} \hat{\mathbf{R}}_z$ and $\hat{\mathbf{S}}_X = -(-1)^c \cdot \mathbf{I}_n + \sum_{z \in CF_X} \hat{\mathbf{S}}_z$. Obviously, we have that $\hat{\mathbf{S}}_X = \mathbf{0}$ if and only if $z^* \in CF_X$, otherwise $\hat{\mathbf{S}}_X = -(-1)^c \cdot \mathbf{I}_n$. By the property of the cover-free sets, there is at least one element in $CF_Y \subseteq [N]$ that is not included in the union set $\cup_{X \in \mathcal{S}} CF_X$ for any $\mathcal{S} = \{X_1, \ldots, X_v\}$ and $Y \notin \mathcal{S}$. Thus, if z^* is randomly chosen and is statistically hidden in the key $K = (\mathbf{A}, \{\mathbf{A}_i\}_{i \in \{0,\ldots,\mu-1\}})$, we have the probability that $\mathrm{H}_K(X_i) = \hat{\mathbf{A}}\hat{\mathbf{R}}_{X_i} - (-1)^c \cdot \mathbf{G}$ for all $X_i \in \mathcal{S}$ and $\mathrm{H}_K(Y) = \hat{\mathbf{A}}\hat{\mathbf{R}}_Y$, is at least $1/N = \frac{1}{O(v^2\ell)}$. This gives a $(1, v, \beta)$-PHF for any arbitrarily chosen but prior fixed $v = \mathsf{poly}(\kappa)$ and some polynomially bounded $\beta = \mathsf{poly}(\kappa)$.

Type-III PHF Construction. The above two types of PHF constructions can be instantiated on either the integer ring $\mathcal{R} = \mathbb{Z}$ for general lattices or the polynomial ring $\mathcal{R} = R = \mathbb{Z}[x]/(x^n + 1)$ for ideal lattices. Our Type-III PHF construction is specific on polynomial ring $\mathcal{R} = R$. Recall that the core idea of our Type-II construction is to homomorphically evaluate a comparison function over $[N]$ (recall that $\hat{\mathbf{S}}_z = \prod_{i=0}^{\mu-1}(1 - b_i^* - b_i) \cdot \mathbf{I}_n$), which in turn is realized by multiplying $\mu = \lceil \log_2 N \rceil$ bit-comparison functions

$$f(b_i^*, b_i) = 1 - b_i^* - b_i = \begin{cases} 0, & \text{if } b_i^* \neq b_i \\ \pm 1, & \text{otherwise} \end{cases}$$

for all $i \in [\log_2 N]$. The use of bit-comparison functions is simply because we want to ensure the magnitude of $s_1(\mathbf{R}_X)$ for all input X is upper bounded by some small parameter β. Since we need one matrix to encode each b_i^* (namely, $\mathbf{A}_i = \hat{\mathbf{A}}\mathbf{R}_i + (1 - b_i^*) \cdot \mathbf{G}$) in the keys, this immediately leads to at least $\log_2 N$ matrices in the Type-II PHF keys. We now show how to obtain a PHF with constant keys by adapting the techniques in [1]. Specifically, as shown in [1], we can cheaply evaluate a comparison function over $[2n]$ in ring $\mathcal{R}_q = \mathbb{Z}_q[x]/(x^n + 1)$ with n being a power of 2 and odd integer q. In more detail, for any $a, b \in [2n]$, we have that

$$f(a, b) = \frac{1}{2n} \sum_{i=0}^{2n-1} (x^{a-b})^i = \begin{cases} 0, & \text{if } a \neq b \\ 1, & \text{otherwise} \end{cases}$$

By replacing the underlying bit-comparison function in our Type-II PHF construction with the above comparison function $f(a, b) \in R_q$ for any $a, b \in [2n]$, we immediately obtain a Type-III PHF construction which reduces the number of matrices in the keys from $\log_2 N$ to $\log_{2n} N$, and thus has constant keys for any $N = \mathsf{poly}(n)$. The downside is that the parameter β for the Type-III PHF is at least n times larger than that of the Type-II PHF. However, for the setting of $N \leq n^2$, we also present an improved Type-III PHF construction with constant keys by using a different way to homomorphically evaluate the same comparison function, so that the resulting parameter β is asymptotically equal to that of our Type-II PHF construction.

1.4 Short Signatures

We now outline the idea on how to construct a generic signature scheme $\mathcal{SIG}$ from lattice-based PHFs in the standard model. Let $n, \bar{m}, m', \ell, q$ be some positive integers, and let $m = \bar{m} + m'$. Given a lattice-based PHF $\mathcal{H} = \{\mathrm{H}_K\}$ from $\{0,1\}^\ell$ to $\mathcal{R}_q^{n \times m'}$, let $\mathbf{B} \in \mathcal{R}_q^{n \times m'}$ be a trapdoor matrix that is compatible with $\mathcal{H}$. Then, the verification key of the generic signature scheme $\mathcal{SIG}$ consists of a uniformly distributed (trapdoor) matrix $\mathbf{A} \in \mathcal{R}_q^{n \times \bar{m}}$, a uniformly random vector $\mathbf{u} \in \mathcal{R}_q^n$, and a random key K for $\mathcal{H}$, i.e., $vk = (\mathbf{A}, \mathbf{u}, K)$. The signing key is a trapdoor $\mathbf{R}$ of $\mathbf{A}$ that allows to sample short vector $\mathbf{e}$ satisfying $\mathbf{A}\mathbf{e} = \mathbf{v}$ for any vector $\mathbf{v} \in \mathcal{R}_q^n$. Given a message $M \in \{0,1\}^\ell$, the signing algorithm first computes $\mathbf{A}_M = (\mathbf{A} \| \mathrm{H}_K(M)) \in \mathcal{R}_q^{n \times m}$, and then uses the trapdoor $\mathbf{R}$ to sample a short vector $\mathbf{e} \in \mathcal{R}^m$ satisfying $\mathbf{A}_M \mathbf{e} = \mathbf{u}$ by employing the sampling algorithms in [31,16,47]. Finally, it returns $\sigma = \mathbf{e}$ as the signature on the message M. The verifier accepts $\sigma = \mathbf{e}$ as a valid signature on M if and only if $\mathbf{e}$ is short and $\mathbf{A}_M \mathbf{e} = \mathbf{u}$. The correctness of the generic scheme $\mathcal{SIG}$ is guaranteed by the nice properties of the sampling algorithms in [31,47].

In addition, if $\mathcal{H} = \{\mathrm{H}_K\}$ is a $(1, \mathsf{poly}, \beta)$-PHF for some real β, we can show that under the ISIS assumption over $\mathcal{R}$, the above signature scheme $\mathcal{SIG}$

is existentially unforgeable against chosen message attacks (EUF-CMA) in the standard model as long as the forger $\mathcal{F}$ makes at most polynomial number $Q = \mathsf{poly}(n)$ of signing queries. Intuitively, given an ISIS challenge instance $(\hat{\mathbf{A}}, \hat{\mathbf{u}})$ in the security reduction, the challenger first generates a trapdoor mode key K' for $\mathcal{H}$ by using $(\hat{\mathbf{A}}, \mathbf{B})$. Then, it defines $vk = (\hat{\mathbf{A}}, \hat{\mathbf{u}}, K')$ and keeps the trapdoor td of K' private. For message M_i in the i-th signing query, we have $\mathbf{A}_{M_i} = (\hat{\mathbf{A}} \| \mathrm{H}_{K'}(M_i)) = (\hat{\mathbf{A}} \| \hat{\mathbf{A}}\mathbf{R}_{M_i} + \mathbf{S}_{M_i}\mathbf{B}) \in \mathcal{R}_q^{n \times m}$. By the programmability of $\mathcal{H}$, with a certain probability we have that $\mathbf{S}_{M_i}$ is invertible for all the Q signing messages $\{M_i\}_{i \in \{1,\dots,Q\}}$, but $\mathbf{S}_{M^*} = \mathbf{0}$ for the forged message M^*. In this case, the challenger can use $\mathbf{R}_{M_i}$ to perfectly answer the signing queries, and use the forged message-signature pair (M^*, σ^*) to solve the ISIS problem by the equation $\mathbf{u} = \mathbf{A}_{M^*}\sigma^* = \hat{\mathbf{A}}(\mathbf{I}_{\bar{m}} \| \mathbf{R}_{M^*})\sigma^*$.

Each signature in the generic scheme $\mathcal{SIG}$ only has a single vector, which is as short as that in [13,47]. In fact, our generic scheme $\mathcal{SIG}$ encompasses the two signature schemes from [13,47] in the sense that both schemes can be seen as the instantiations of $\mathcal{SIG}$ using the Type-I PHF construction. Due to the inefficiency of the concrete PHFs, both schemes [13,47] had large verification keys consisting of a linear number of matrices. By instantiating $\mathcal{SIG}$ with our efficient Type-II and Type-III PHF constructions, we can obtain concrete signatures $\mathcal{SIG}_1$ with logarithmic verification keys on general lattices and $\mathcal{SIG}_2$ with constant verification keys on ideal lattices, respectively. But because we can only show that our Type-II and Type-III PHF constructions are $(1, v, \beta)$-PHFs for some arbitrary but prior fixed polynomial $v = \mathsf{poly}(n)$, both $\mathcal{SIG}_1$ and $\mathcal{SIG}_2$ are only provably secure in the q-bounded EUF-CMA security model (i.e., EUF-qCMA), which requires the number Q of the adversary's signing queries is known in advance (so that we can set a polynomial $v \geq Q$ for the security proof). Moreover, both $\mathcal{SIG}_1$ and $\mathcal{SIG}_2$ have security proofs with reduction loss about nQ^2, and the parameter $\bar{\beta}$ for the underlying ISIS assumption contains a factor of Q^2. By carefully combining our Type-II PHF with a simple weak Type-I PHF and introducing a very short tag to each signature, we can obtain an improved short signature scheme $\mathcal{SIG}_3$ with logarithmic verification keys in the standard model, which further cuts down the reduction loss by a factor of Q and the parameter $\bar{\beta}$ for the underlying ISIS assumption by a factor of Q^2. However, $\mathcal{SIG}_3$ also shares the same limitation as $\mathcal{SIG}_1$ and $\mathcal{SIG}_2$: it is also only provably secure in the EUF-qCMA model.

In order to remove the above limitation, we further construct two new signatures $\mathcal{SIG}_4$ and $\mathcal{SIG}_5$, which are provably secure in the standard EUF-CMA model, and achieve the same asymptotically reduction loss $Q \cdot \tilde{O}(n)$ and hardness parameter $\bar{\beta} = \tilde{O}(n^{5.5})$ as $\mathcal{SIG}_3$.[4] At a high level, we basically replace the weak Type-I PHF in $\mathcal{SIG}_3$ with a set of weak PHFs to realize the confined guessing technique [9] so that a tag space with super-polynomial size can be used to handle all polynomially bounded $Q = \mathsf{poly}(n)$ (note that the size of the tag space in $\mathcal{SIG}_3$ is required to be slightly larger than Q for a tighter reduction, and thus can only be determined after knowing Q), while at the same time achieving the

[4] We write $f(n) = \tilde{O}(g(n))$ if $f(n) = O(g(n) \cdot \log^c(n))$ for some constant c.

same asymptotic reduction loss as $\mathcal{SIG}_3$. More concretely, we will reuse the subroutines of our Type-II and Type-III PHFs to compute a set of index functions corresponding to the length of the sub-tags used in generating a signature so that we can dynamically choose an appropriate guessing space for the challenge tag in the security proof. The main reason that we can obtain a much better reduction loss than that of [9] is because our Type-II and Type-III PHFs essentially imply the existence of tag-based signatures supporting any polynomially large (but prior fixed) tag-collision parameter v, say, $v = \omega(\log n)$. In particular, this large tag-collision parameter v allows us to choose a small guessing space for the challenge tag in the security proof, which immediately gives a large guessing probability (and thus a small overall reduction loss). In contrast, the lattice-based signature in [9] only supports a very small tag-collision parameter $v = 1$, which requires a more complex strategy to choose a much larger guessing space for the challenge tag in the security proof. We note that our improvement on the reduction loss is not obtained without a price: the computational time of our signatures $\mathcal{SIG}_4$ and $\mathcal{SIG}_5$ is a linear function of v.

In Table 1, we give a (rough) comparison of lattice-based signature schemes in the standard model. For simplicity, the message length is set to be n. Let constant $c > 1$ and $d = O(\log_c n)$ be the parameters for the use of the confined guessing technique in [9,26,6]. We compare the size of verification keys and signatures in terms of the number of "basic" elements as in [26,6]. On general lattices, the "basic" element in the verification keys is a matrix over $\mathbb{Z}_q$ whose size is mainly determined by the underlying hard lattices, while the "basic" element in the signatures is a lattice vector. On ideal lattices, the "basic" element in the verification keys can be represented by a vector. Almost all schemes on general lattices such as [16,13,47,9,6] can be instantiated from ideal lattices, and thus roughly saves a factor of n in the verification key size. However, the schemes marked with '*' from ideal lattices have no realizations over general lattices. We ignore the constant factors in the table to avoid clutter. Since all schemes only have a single "basic" element in the signing keys, we also omit the corresponding comparison in the size of signing keys for succinctness. Finally, we note that the signature scheme in [47] (marked with '†') is essentially identical to the one in [13] except that an improved security reduction under a weaker assumption was provided. As shown in Table 1, the scheme in [6] only has a constant number of "basic" elements in the verification key. However, because a large (I)SIS parameter $\bar{\beta} = \tilde{O}(d^{2d} \cdot n^{5.5})$ is needed (which requires a super-polynomial modulus $q > \bar{\beta}$), the actual bit size to represent each "basic" element in [6] is at least $O(d) = O(\log n)$ times larger than that in [26] and our scheme $\mathcal{SIG}_4$ (we also note that the modulus q for our $\mathcal{SIG}_1$ and $\mathcal{SIG}_2^*$ is also very large as the corresponding ISIS parameter $\bar{\beta}$ depends on Q^2). Even if we do not take account of the reduction loss, the bit size of the verification key in [6] is already as large as that in [26] and our scheme $\mathcal{SIG}_4$. Moreover, compared to all known standard model signature schemes, our scheme $\mathcal{SIG}_5$ simultaneously achieves constant verification keys, short signatures and the tightest reduction.

Table 1. Rough comparison of lattice-based signatures in the standard model (Since all schemes only have a single "basic" element in the signing keys, we also omit the corresponding comparison in the size of signing keys for succinctness. The reduction loss is the ratio ϵ/ϵ' between the success probability ϵ of the forger and the success probability ϵ' of the reduction. Real $\bar{\beta}$ is the parameter for the (I)SIS problem, and "Security" denotes the security notion that is achieved by the corresponding schemes. Constant $c > 1$ and $d = O(\log_c n)$ are the parameters in [9,26,6]. The scheme in [14] requires the existence of a PRF that can be computed by a NC^1 circuit with input length $\ell = \mathsf{poly}(n)$ and depth $c' \log \ell$ for some $c' > 1$. The constant parameter r can be any integer $r \geq 1$ for our scheme $\mathcal{SIG}_2^*$. Since the (I)SIS parameter $\bar{\beta}$ for our $\mathcal{SIG}_1$ and $\mathcal{SIG}_2^*$ is a quadratic function of Q, we need to use a large modulus $q > \bar{\beta}$)

Schemes	Verification key	Signature	Reduction loss	(I)SIS param $\bar{\beta}$	Security
LM08 [45] *	1	$\log n$	Q	$\tilde{O}(n^2)$	EUF-CMA
CHKP10 [16]	n	$\log n$	Q	$\tilde{O}(n^{1.5})$	
Boyen10 [13]	n	1	Q	$\tilde{O}(n^{3.5})$	
MP12 [47] †	n	1	Q	$\tilde{O}(n^{2.5})$	
BHJ$^+$14 [9]	1	d	$(Q^2/\epsilon)^c$	$\tilde{O}(n^{2.5})$	
DM14 [26] *	d	1	$(Q^2/\epsilon)^c$	$\tilde{O}(n^{3.5})$	
AS15 [6]	1	1	$(Q^2/\epsilon)^c$	$\tilde{O}(d^{2d} \cdot n^{5.5})$	
BL16 [14]	n	1	n	$\tilde{O}(\ell^{4c'} \cdot n^{3.5})$	
KONT20 [39] *	d, n	1	$(Q/n)^c, Q/n$	$\tilde{O}(n^{3.5})$	
Our $\mathcal{SIG}_1$	$\log n$	1	$n \cdot Q^2$	$Q^2 \cdot \tilde{O}(n^{5.5})$	EUF-qCMA
Our $\mathcal{SIG}_2^*$	r	1	$n \cdot Q^2$	$Q^2 \cdot \tilde{O}(n^{7.5+2/r})$	
Our $\mathcal{SIG}_3$	$\log n$	1	$Q \cdot \tilde{O}(n)$	$\tilde{O}(n^{5.5})$	
Our $\mathcal{SIG}_4$	$\log n$	1	$Q \cdot \tilde{O}(n)$	$\tilde{O}(n^{5.5})$	EUF-CMA
Our $\mathcal{SIG}_5^*$	1	1	$Q \cdot \tilde{O}(n)$	$\tilde{O}(n^{5.5})$	

1.5 Identity-based Encryption

At STOC 2008, Gentry et al. [31] constructed a variant of the LWE-based public-key encryption (PKE) scheme [51]. Informally, the public key of their scheme [31] contained a matrix $\mathbf{A}$ and a vector $\mathbf{u}$, and the secret key was a short vector $\mathbf{e}$ satisfying $\mathbf{Ae} = \mathbf{u}$. Recall that in our generic signature scheme $\mathcal{SIG}$, any valid message-signature pair (M, σ) under the verification key $vk = (\mathbf{A}, \mathbf{u}, K)$ also satisfies an equation $\mathbf{A}_M \sigma = \mathbf{u}$, where $\mathbf{A}_M = (\mathbf{A} \| \mathrm{H}_K(M))$. This observation allows to give a generic IBE scheme $\mathcal{IBE}$ from lattice-based PHFs by combining our generic signature scheme $\mathcal{SIG}$ with the PKE scheme in [31]. Concretely, let the master public key mpk and the master secret key msk of the IBE system be the verification key vk and the secret signing key sk of $\mathcal{SIG}$, respectively, i.e., $(mpk, msk) = (vk, sk)$. Then, for each identity id, we simply generate a "signature" $sk_{id} = \sigma$ on id under the master public key mpk as the user private key, i.e., $\mathbf{A}_{id} sk_{id} = \mathbf{u}$, where $\mathbf{A}_{id} = (\mathbf{A} \| \mathrm{H}_K(id))$. Finally, we run the encryption

Table 2. Rough Comparison of lattice-based IBEs in the standard model (Since all the schemes only have a single "basic" element in both the master secret key and the user private key, we omit them in the comparison for succinctness. The reduction loss is the ratio ϵ/ϵ' between the success probability ϵ of the attacker and the success probability ϵ' of the reduction. Real α is the parameter for the LWE problem, and "security" standards for the corresponding security model for security proofs. The scheme in [14] requires the existence of a PRF that can be computed by a NC^1 circuit with input length $\ell = \mathsf{poly}(n)$ and depth $c' \log \ell$ for some $c' > 1$. $\mathsf{poly}(n)$ (resp. $\mathsf{superpoly}(n)$) represents fixed but large polynomial (resp. super-polynomial). The parameter $t = \mathsf{poly}(n) > Q$ denotes the running time of the adversary in [38]. The constant r satisfies $n^{1/r} > 3 + r$ in [1]. The constant r' can be any integer $r' \geq 1$ for our scheme $\mathcal{IBE}_2^*$)

Schemes	Master public key	Ciphertext	Reduction loss	LWE param $1/\alpha$	Security
ABB10a [3]	n^3	n^2	1	$\tilde{O}(n^{2n})$	IND-sID-CPA
ABB10b [2]	n	1	Q	$\tilde{O}(n^2)$	IND-ID-CPA
CHKP10 [16]	n	n	Q^2	$\tilde{O}(n^{1.5})$	
Yamada16 [56]	$n^{1/c}$	1	$n(Q/\epsilon)^c$	$n^{\omega(1)}$	
BL16 [14]	n	1	n	$\mathsf{superpoly}(n)$	
KY16 [40]	$n^{1/c}$	1	$Q^{c^2+c}/(n^{c^2-1}\epsilon^{c^2+c-1})$	$n^{2.5+2\mu}$	
Yamada17 [57]	$(\log n)^2$	1	$n^2 \cdot Q/\epsilon$	$\mathsf{poly}(n)$	
JKN21 [38]	$\log n$	1	$\tilde{O}(n^6)$	t^2/ϵ	
ALWW21 [1]	$\omega(1)$	1	$n^{1/r} \cdot Q$	$\tilde{O}(n^{7.5+4/r})$	
Our $\mathcal{IBE}_1$	$\log n$	1	$n \cdot Q^2$	$Q^2 \cdot \tilde{O}(n^{6.5})$	IND-qID-CPA
Our $\mathcal{IBE}_2^*$	r'	1	$n \cdot Q^2$	$Q^2 \cdot \tilde{O}(n^{8.5+2/r'})$	

algorithm of [31] with "public key" $(\mathbf{A}_{id}, \mathbf{u})$ as a sub-routine to encrypt plaintexts under the identity id. It is easy to show that the above construction is correct.

The problem is how to rely the security of the above IBE scheme on the LWE assumption. In the conference version [60], we resort to PHFs with an enhanced property called high min-entropy, which basically requires that the properties of PHFs still hold if some information related to the trapdoor is leaked. In this version, we improve the generic IBE scheme $\mathcal{IBE}$ from lattice-based PHFs, which removes the requirement of the high min-entropy property, and makes it easier to be instantiated from ideal lattices (because the high min-entropy property in [60] is obtained by applying the generalized leftover hash lemma [23] which is typically not applicable on general polynomial rings). By instantiating $\mathcal{IBE}$ with our Type-II and Type-III PHF constructions, we immediately obtain standard model IBE schemes $\mathcal{IBE}_1$ with logarithmic master public keys on general lattices and $\mathcal{IBE}_2$ with constant master public keys on ideal lattices, respectively. Similarly for our signatures from generic constructions, both $\mathcal{IBE}_1$ and $\mathcal{IBE}_2$ is only provably secure in the q-bounded security model, namely, IND-qID-CPA. Although one can remove this dependence on Q by setting a default super-polynomial Q, e.g., $Q = 2^{\omega(\log n)}$, the computational time of the resulting schemes will also be a linear function of such Q.

In Table 2, we give a (rough) comparison of lattice-based IBEs in the standard model. For simplicity, the identity length is set to be n. (Note that one can use a collision-resistant hash function with output length n to deal with identities with arbitrary length.) Similarly, we compare the size of master public keys and ciphertexts in terms of the number of "basic" elements. On general lattices, the "basic" element in the master public keys is a matrix, while the "basic" element in the ciphertexts is a vector. If instantiated from ideal lattices, the "basic" element in the master public keys can be represented by a vector, and thus roughly saves a factor of n in the master public key size. We ignore the constant factor in the table to avoid clutter. Compared to other fully secure IBE schemes in the standard model, our schemes $\mathcal{IBE}_1$ and $\mathcal{IBE}_2$ have very short master public key. However, such an improvement is not obtained without a penalty: in addition to the IND-qID-CPA security model, the security loss and the hardness parameter for the underlying (ring-)LWE assumptions also depend on the maximum number Q of the adversary's user key extraction queries.

Since both the improvement and the downside are inherited from our concrete Type-II and Type-III PHF constructions, this situation can be immediately changed if one can find a better lattice-based PHF. In particular, it is worth to note that after the publication of our conference paper, several works [57,38,1] had constructed much more efficient IBEs with shorter master public keys in the full security model by implicitly constructing lattice-based PHFs with nice features and shorter parameters.

1.6 Other Related Work

Hofheinz and Kiltz [35] first introduced the notion of PHF based on group hash functions, and gave a concrete $(2, 1)$-PHF instantiation. Then, the work [34] constructed a $(u, 1)$-PHF for any $u \geq 1$ by using cover-free sets. Later, Yamada et al. [58] reduced the key size from $O(u^2\ell)$ in [34] to $O(u\sqrt{\ell})$ by combining the two-dimensional representation of cover-free sets with the bilinear groups, where ℓ was the bit size of the inputs. At CRYPTO 2012, Hanaoka et al. [33] showed that it was impossible to construct *algebraic* $(u, 1)$-PHF over prime order groups in a black-box way such that its key has less than u group elements.[5] Later, Freire et al. [28] got around the impossibility result of [33] and constructed a $(\mathsf{poly}, 1)$-PHF by adapting PHFs to the multilinear maps setting. Despite its great theoretical interests, the current state of multilinear maps might be a big obstacle in any attempt to securely and efficiently instantiate the PHFs in [28]. More recently, Catalano et al. [17] introduced a variant of traditional PHF called asymmetric PHF over bilinear maps, and used it to construct (homomorphic) signature schemes with short verification keys.

All the above PHF constructions [35,34,58,28,17] seem specific to groups with nice properties, which might constitute a main barrier to instantiate them from lattices. Although several lattice-based schemes [2,16] had employed a similar partitioning proof trick as that was captured by the traditional PHFs, it was still

[5] Informally, an algorithm is algebraic if there is a way to compute the representation of a group element output by the algorithm in terms of its input group elements [12].

an open problem to formalize and construct PHFs from lattices [36]. We put forward this study by introducing the lattice-based PHF and demonstrate its power in constructing lattice-based signatures and IBEs in the standard model. Our PHFs also provide a modular way to investigate several existing cryptographic constructions from lattices [2,13,47].

2 Preliminaries

2.1 Notation

Let κ be the natural security parameter, and all other quantities are implicitly dependent on κ. The function $\log_c$ denotes the logarithm with base c, and we use log to denote the natural logarithm. The standard notation O, ω are used to classify the growth of functions. If $f(n) = O(g(n) \cdot \log^c(n))$ for some constant c, we write $f(n) = \tilde{O}(g(n))$. By $\mathsf{poly}(n)$ we denote an arbitrary function $f(n) = O(n^c)$ for some constant c. A function $f(n)$ is negligible in n if for every positive c, we have $f(n) < n^{-c}$ for sufficiently large n. By $\mathrm{negl}(n)$ we denote an arbitrary negligible function. A probability is said to be overwhelming if it is $1 - \mathrm{negl}(n)$. The notation $\leftarrow$ denotes randomly choosing elements from some distribution (or the uniform distribution over some finite set). If a random variable x follows some distribution D, we denote it by $x \backsim D$.

By $\mathbb{R}$ (resp. $\mathbb{Z}$) we denote the set of real numbers (resp. integers). For any positive $N \in \mathbb{Z}$, the notation $[N]$ denotes the set $\{0, 1, \ldots, N-1\}$. Vectors are used in the column form and denoted by bold lower-case letters (e.g., $\mathbf{x}$). Matrices are treated as the sets of column vectors and denoted by bold capital letters (e.g., $\mathbf{X}$). The concatenation of the columns of $\mathbf{X} \in \mathbb{R}^{n\times m}$ followed by the columns of $\mathbf{Y} \in \mathbb{R}^{n\times m'}$ is denoted as $(\mathbf{X}\|\mathbf{Y}) \in \mathbb{R}^{n\times(m+m')}$. For any element $0 \le v \le q$ and integer $b \ge 2$, we denote $\mathsf{BitDecomp}_{q,b}(v)$ as the k-dimensional bit-decomposition of v in base b, where $k = \lceil \log_b q \rceil$. We usually omit the subscript b in $\mathsf{BitDecomp}_{q,b}$ if $b = 2$ for simplicity. By $\|\cdot\|$ and $\|\cdot\|_\infty$ we denote the l_2 and l_∞ norm, respectively. The norm of a matrix $\mathbf{X}$ is defined as the norm of its longest column (i.e., $\|\mathbf{X}\| = \max_i \|\mathbf{x}_i\|$). The largest singular value of a matrix $\mathbf{X}$ is $s_1(\mathbf{X}) = \max_{\mathbf{u}} \|\mathbf{X}\mathbf{u}\|$, where the maximum is taken over all unit vectors $\mathbf{u}$.

We say that a hash function $H : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$ is an encoding with full-rank differences (FRD) if the following two conditions hold: 1) for any $\mathbf{u} \neq \mathbf{v}$, the matrix $H(\mathbf{u}) - H(\mathbf{v}) \in \mathbb{Z}_q^{n\times n}$ is invertible over $\mathbb{Z}_q^{n\times n}$; and 2) H is computable in polynomial time in $n \log q$. As shown in [2,22], FRD encodings supporting the exponential size domain $\mathbb{Z}_q^n$ can be efficiently constructed. We say that set S does not cover set T if there exists at least one element $t \in T$ such that $t \notin S$. Let $CF = \{CF_X\}_{X\in[L]}$ be a family of subsets of $[N]$. The family CF is said to be v-cover-free over $[N]$ if for any subset $\mathcal{S} \subseteq [L]$ of size at most v, then the union $\cup_{X\in\mathcal{S}} CF_X$ does not cover CF_Y for all $Y \notin \mathcal{S}$. Besides, we say that CF is η-uniform if every subset CF_X in the family $CF = \{CF_X\}_{X\in[L]}$ have size $\eta \in \mathbb{Z}$. Furthermore, there exists an efficient algorithm to generate cover-free sets [27,43].

Lemma 1. *There is a deterministic polynomial time algorithm that takes two integers $L = 2^\ell$ and $v \in \mathbb{Z}$ as inputs, returns an η-uniform, v-cover-free sets $CF = \{CF_X\}_{X \in [L]}$ over some $[N]$, where $N \leq 16v^2\ell$ and $\eta = N/4v$.*

2.2 Lattices and Gaussian Distributions

An m-dimensional full-rank lattice $\mathbf{\Lambda} \subset \mathbb{R}^m$ is the set of all integral combinations of m linearly independent vectors $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_m) \in \mathbb{R}^{m \times m}$, i.e., $\mathbf{\Lambda} = \mathcal{L}(\mathbf{B}) = \{\sum_{i=1}^m x_i \mathbf{b}_i : x_i \in \mathbb{Z}\}$. For $\mathbf{x} \in \mathbf{\Lambda}$, define the Gaussian function $\rho_{s,\mathbf{c}}(\mathbf{x})$ over $\mathbf{\Lambda} \subseteq \mathbb{Z}^m$ centered at $\mathbf{c} \in \mathbb{R}^m$ with parameter $s > 0$ as $\rho_{s,\mathbf{c}}(\mathbf{x}) = \exp(-\pi\|\mathbf{x} - \mathbf{c}\|^2/s^2)$. Let $\rho_{s,\mathbf{c}}(\mathbf{\Lambda}) = \sum_{\mathbf{x} \in \mathbf{\Lambda}} \rho_{s,\mathbf{c}}(\mathbf{x})$, and define the discrete Gaussian distribution over $\mathbf{\Lambda}$ as $D_{\mathbf{\Lambda},s,\mathbf{c}}(\mathbf{y}) = \frac{\rho_{s,\mathbf{c}}(\mathbf{y})}{\rho_{s,\mathbf{c}}(\mathbf{\Lambda})}$, where $\mathbf{y} \in \mathbf{\Lambda}$. The subscripts s and $\mathbf{c}$ are taken to be 1 and $\mathbf{0}$ (resp.) when omitted. The following result was proved in [48,31,50].

Lemma 2. *For any positive integer $m \in \mathbb{Z}$, vector $\mathbf{y} \in \mathbb{Z}^m$ and large enough $s \geq \omega(\sqrt{\log m})$, we have that*

$$\Pr_{\mathbf{x} \leftarrow D_{\mathbb{Z}^m,s}}[\|\mathbf{x}\| > s\sqrt{m}] \leq 2^{-m} \text{ and } \Pr_{\mathbf{x} \leftarrow D_{\mathbb{Z}^m,s}}[\mathbf{x} = \mathbf{y}] \leq 2^{1-m}.$$

Following [47,26], we say that a random variable X over $\mathbb{R}$ is subgaussian with parameter s if for all $t \in \mathbb{R}$, the (scaled) moment-generating function satisfies $\mathbb{E}(\exp(2\pi t X)) \leq \exp(\pi s^2 t^2)$. If X is subgaussian, then its tails are dominated by a Gaussian of parameter s, i.e., $\Pr[|X| \geq t] \leq 2\exp(-\pi t^2/s^2)$ for all $t \geq 0$. As a special case, any B-bounded symmetric random variable X (i.e., $|X| \leq B$ always) is subgaussian with parameter $B\sqrt{2\pi}$. Besides, we say that a random matrix $\mathbf{X}$ is subgaussian with parameter s if all its one-dimensional marginals $\mathbf{u}^T\mathbf{X}\mathbf{v}$ for unit vectors $\mathbf{u}, \mathbf{v}$ are subgaussian with parameter s. In such a definition, the concatenation of independent subgaussian vectors with parameter s, interpreted either as a vector or as a matrix, is subgaussian with parameter s. In particular, the distribution $D_{\mathbf{\Lambda},s}$ for any lattice $\mathbf{\Lambda} \subset \mathbb{R}^n$ and $s > 0$ is subgaussian with parameter s. For random subgaussian matrix, we have the following result from the non-asymptotic theory of random matrices [53].

Lemma 3. *Let $\mathbf{X} \in \mathbb{R}^{n \times m}$ be a random subgaussian matrix with parameter s. There exists a universal constant $C \approx 1/\sqrt{2\pi}$ such that for any $t \geq 0$, we have $s_1(\mathbf{X}) \leq C \cdot s \cdot (\sqrt{m} + \sqrt{n} + t)$ except with probability at most $2\exp(-\pi t^2)$.*

We also have the following useful lemma from [40], which basically says that we can re-randomize an unknown Gaussian variable.

Lemma 4 ([40]). *Let $n, m, q > 0$ be integers. Let $\mathbf{b} \in \mathbb{Z}_q^m$ be arbitrary and $\mathbf{x} \leftarrow D_{\mathbb{Z}^m,r}$ with $r \geq \max\{\omega(\sqrt{\log n}), \omega(\sqrt{\log m})\}$. Then, for any matrix $\mathbf{R} \in \mathbb{Z}^{n \times m}$ and real $\sigma > s_1(\mathbf{R})$, there exists a PPT algorithm $\mathsf{ReRand}_{\mathbb{Z}}(\mathbf{R}, \mathbf{b} + \mathbf{x}, r, \sigma)$ that outputs $\mathbf{b}' = \mathbf{R}\mathbf{b} + \mathbf{x}' \in \mathbb{Z}_q^n$ where $\mathbf{x}'$ is distributed statistically close to $D_{\mathbb{Z}^n,2r\sigma}$.*

Let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ be a matrix for some positive $n, m, q \in \mathbb{Z}$, consider the following two lattices: $\mathbf{\Lambda}_q^\perp(\mathbf{A}) = \{\mathbf{e} \in \mathbb{Z}^m \;\; s.t.\; \mathbf{A}\mathbf{e} = 0 \mod q\}$ and $\mathbf{\Lambda}_q(\mathbf{A}) =$

$\{\mathbf{y} \in \mathbb{Z}^m \ s.t. \ \exists \mathbf{s} \in \mathbb{Z}^n, \ \mathbf{A}^T\mathbf{s} = \mathbf{y} \mod q\}$. By definition, we have $\mathbf{\Lambda}_q^\perp(\mathbf{A}) = \mathbf{\Lambda}_q^\perp(\mathbf{CA})$ for any invertible $\mathbf{C} \in \mathbb{Z}_q^{n\times n}$. We have several useful facts for Gaussian distributions from [50,31,47].

Lemma 5. *For any positive integer n, prime $q > 2$, sufficiently large $m = O(n \log q)$ and real $s \geq \omega(\sqrt{\log m})$, we have that for a uniformly random matrix $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$, the following facts hold:*

- *for variable $\mathbf{e} \sim D_{\mathbb{Z}^m,s}$, the distribution of $\mathbf{u} = \mathbf{Ae} \mod q$ is statistically close to uniform over $\mathbb{Z}_q^n$;*
- *for any $\mathbf{c} \in \mathbb{R}^m$ and every $\mathbf{y} \in \mathbf{\Lambda}_q^\perp(\mathbf{A})$, $\Pr_{\mathbf{x}\leftarrow D_{\mathbf{\Lambda}_q^\perp(\mathbf{A}),s,\mathbf{c}}}[\mathbf{x} = \mathbf{y}] \leq 2^{1-m}$;*
- *for any fixed $\mathbf{u} \in \mathbb{Z}_q^n$ and arbitrary $\mathbf{v} \in \mathbb{R}^m$ satisfying $\mathbf{Av} = \mathbf{u} \mod q$, the conditional distribution of $\mathbf{e} \sim D_{\mathbb{Z}^m,s}$ given $\mathbf{Ae} = \mathbf{u} \mod q$ is exactly $\mathbf{v} + D_{\mathbf{\Lambda}_q^\perp(\mathbf{A}),s,-\mathbf{v}}$.*

Trapdoors. In 1999, Ajtai [5] proposed the first trapdoor generation algorithm to output an essentially uniform trapdoor matrix $\mathbf{A}$ that allows to efficiently sample short vectors in $\mathbf{\Lambda}_q^\perp(\mathbf{A})$. This trapdoor generation algorithm had been improved in [47]. Let $\mathbf{I}_n$ be the $n \times n$ identity matrix. We now recall the publicly known trapdoor matrix $\mathbf{G}_b$ in [47]. Formally, for any prime $q > 2$, integers $n \geq 1, b \geq 2$ and $k = \lceil \log_b q \rceil$, define $\mathbf{G}_b = (\mathbf{I}_n, b\mathbf{I}_n, \ldots, b^{k-1}\mathbf{I}_n) \in \mathbb{Z}_q^{n\times nk}$.[6] We usually omit the subscript b in $\mathbf{G}_b$ if $b = 2$. Then, the lattice $\mathbf{\Lambda}_q^\perp(\mathbf{G})$ has a publicly known short basis $\mathbf{T}_b \in \mathbb{Z}^{nk\times nk}$ with $\|\mathbf{T}_b\| \leq \max\{\sqrt{b^2+1}, \sqrt{k}\}$. For any vector $\mathbf{u} \in \mathbb{Z}_q^n$, the basis $\mathbf{T}_b \in \mathbb{Z}_q^{nk\times nk}$ can be used to sample short vector $\mathbf{e} \sim D_{\mathbb{Z}^{nk},s}$ satisfying $\mathbf{G}_b\mathbf{e} = \mathbf{u}$ for any $s \geq \omega(\sqrt{\log n})$ in quasi-linear time. Besides, we can also deterministically compute a short vector $\mathbf{v} = \mathbf{G}_b^{-1}(\mathbf{u}) \in \mathbb{Z}_b^{nk}$ such that $\mathbf{G}_b\mathbf{v} = \mathbf{u}$.

Definition 1 (G-trapdoor [47]). *For any integers $n, \bar{m}, q \in \mathbb{Z}, k = \lceil \log_b q \rceil$, and matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times\bar{m}}$, the $\mathbf{G}$-trapdoor for $\mathbf{A}$ is a matrix $\mathbf{R} \in \mathbb{Z}^{(\bar{m}-nk)\times nk}$ such that $\mathbf{A}\begin{bmatrix}\mathbf{R}\\ \mathbf{I}_{nk}\end{bmatrix} = \mathbf{SG}_b$ for some invertible tag $\mathbf{S} \in \mathbb{Z}_q^{n\times n}$. The quality of the trapdoor is measured by its largest singular value $s_1(\mathbf{R})$.*

If $\mathbf{R}$ is a $\mathbf{G}$-trapdoor for $\mathbf{A}$, one can obtain a $\mathbf{G}$-trapdoor $\mathbf{R}'$ for any extension $(\mathbf{A}\|\mathbf{B})$ by padding $\mathbf{R}$ with zero rows. In particular, we have $s_1(\mathbf{R}') = s_1(\mathbf{R})$. Besides, the rows of $\begin{bmatrix}\mathbf{R}\\ \mathbf{I}_{nk}\end{bmatrix}$ in Definition 1 can appear in any order, since this just induces a permutation of $\mathbf{A}$'s columns [47].

Proposition 1 ([47]). *Given any integers $n \geq 1$, $q > 2$, sufficiently large $\bar{m} = O(n \log q)$ and a tag $\mathbf{S} \in \mathbb{Z}_q^{n\times n}$, there is an efficient randomized algorithm $\mathsf{TrapGen}_{\mathbb{Z}}(1^n, 1^{\bar{m}}, q, \mathbf{S})$ that outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times\bar{m}}$ and a $\mathbf{G}$-trapdoor $\mathbf{R} \in \mathbb{Z}_q^{(\bar{m}-nk)\times nk}$ with quality $s_1(\mathbf{R}) \leq \sqrt{\bar{m}} \cdot \omega(\sqrt{\log n})$ such that the distribution of $\mathbf{A}$ is $\mathrm{negl}(n)$-far from uniform and $\mathbf{A}\begin{bmatrix}\mathbf{R}\\ \mathbf{I}_{nk}\end{bmatrix} = \mathbf{SG}_b$, where $k = \lceil \log_b q \rceil$.*

[6] Note that this definition of $\mathbf{G}_b$ is equivalent to $\mathbf{G}_b = \mathbf{I}_k \otimes (1, b, \ldots, b^{k-1})^T$ in [47] under the column permutation.

In addition, given a $\mathbf{G}$*-trapdoor* $\mathbf{R}$ *of* $\mathbf{A} \in \mathbb{Z}_q^{n \times \bar{m}}$ *for some invertible tag* $\mathbf{S} \in \mathbb{Z}_q^{n \times n}$*, any* $\mathbf{U} \in \mathbb{Z}_q^{n \times n'}$ *for some integer* $n' \geq 1$ *and real* $s \geq s_1(\mathbf{R}) \cdot \omega(\sqrt{\log n})$*, there is an algorithm* $\mathsf{SampleD}_{\mathbb{Z}}(\mathbf{R}, \mathbf{A}, \mathbf{S}, \mathbf{U}, s)$ *that samples from a distribution within* $\mathrm{negl}(n)$ *statistical distance of* $\mathbf{E} \sim (D_{\mathbb{Z}^{\bar{m}},s})^{n'}$ *satisfying* $\mathbf{AE} = \mathbf{U}$*.*

2.3 Rings and Trapdoors

Rings. Let n be a power of 2, and define the cyclotomic ring $R = \mathbb{Z}[x]/(x^n + 1)$. For any integer $q > 0$, define $R_q = \mathbb{Z}_q[x]/(x^n + 1)$ analogously. As in [26], we require that $x^n + 1$ does not split into low degree polynomials modulo the prime factors of q.

Lemma 6 ([26]). *Let* $n \geq 4$ *be a power of* 2*,* $q \geq 3$ *a power of* 3*, and* $R_q = \mathbb{Z}_q[x]/(x^n + 1)$*. Then, any nonzero polynomial* $t \in R_q$ *of degree* $d < n/2$ *and coefficients in* $\{$*-1,0,1*$\}$ *is invertible in* R_q*.*

Let $\phi : R \to \mathbb{Z}^n$ be the coefficient embedding that maps a polynomial into its coefficient vector (in the column form). Then, we define the norm of a polynomial $a \in R$ as the (Euclidean) norm of its coefficient vector, i.e., $\|a\| = \|\phi(a)\|$. By identifying R with $\mathbb{Z}^n$ under the map ϕ, the discrete Gaussian distribution over R can be defined as $D_{R,s} = D_{\mathbb{Z}^n,s}$.

Trapdoors over rings. We can also identity $R = \mathbb{Z}[x]/(x^n + 1)$ with the subring of anti-circulant matrices in $\mathbb{Z}^{n \times n}$ by viewing each ring element $a \in R$ as a linear transformation $b \to a \cdot b$ over the coefficient embedding of R, i.e., by treating a as a matrix in $\mathbb{Z}^{n \times n}$ such that the i-th column is the coefficient of $a \cdot x^i \mod f(x)$ where $i \in \{0, \ldots, n-1\}$. Formally, we let $\mathsf{Rot} : R \to \mathbb{Z}^{n \times n}$ be the ring homomorphism that maps a polynomial into its anti-circulant matrix, i.e., $\mathsf{Rot}(a) = (\phi(a), \phi(ax), \ldots, \phi(ax^{n-1})) \in \mathbb{Z}^{n \times n}$. The definition of Rot can be naturally extended to vectors and matrices over R in a coordinate-wise way.

Lemma 7 ([26]). *Let* $n \geq 4$ *be a power of* 2*,* $q \geq 3$ *a power of* 3*, and* $R_q = \mathbb{Z}_q[x]/(x^n + 1)$*. Let* $m, k > 0$ *be integers, and let* $s > 0$ *be a real. Then, if* $\mathbf{R} \leftarrow (D_{R,s})^{m \times k}$*, with overwhelming probability we have* $s_1(\mathbf{R}) = s_1(\mathsf{Rot}(\mathbf{R})) \leq s\sqrt{n} \cdot O(\sqrt{m} + \sqrt{k} + \omega(\sqrt{\log n}))$*.*

We also have the following Corollary of Lemma 4 for the ring setting.

Lemma 8. *Let* $n, w, k, q > 0$ *be integers. Let* $\mathbf{b} \in \mathbb{Z}_q^w$ *be arbitrary and* $\mathbf{x} \leftarrow (D_{R,r})^w$ *with* $r \geq \max\{\omega(\sqrt{\log kn}), \omega(\sqrt{\log wn})\}$*. Then, for any* $\mathbf{R} \in R_q^{k \times w}$ *and real* $\sigma > s_1(\mathbf{R})$*, there exists a PPT algorithm* $\mathsf{ReRand}_R(\mathbf{R}, \mathbf{b}{+}\mathbf{x}, r, \sigma)$ *that outputs* $\mathbf{b}' = \mathbf{Rb} + \mathbf{x}' \in R_q^k$ *where* $\mathbf{x}'$ *is distributed statistically close to* $(D_{R,2r\sigma})^k$*.*

Let $\mathbf{A} \in R_q^{1 \times \bar{m}}$, define two lattices $\mathbf{\Lambda}_q^{\perp}(\mathbf{A}) = \{\mathbf{e} \in R^{\bar{m}} \ \ s.t.\ \mathbf{Ae} = 0\}$ and $\mathbf{\Lambda}_q(\mathbf{A}) = \{\mathbf{y} \in R^{\bar{m}} \ s.t.\ \exists x \in R_q,\ \mathbf{A}^T x = \mathbf{y}\}$. We have that $\mathbf{\Lambda}_q^{\perp}(\mathbf{A})$ and $\mathbf{\Lambda}_q(\mathbf{A})$ over R are equivalent to $\mathbf{\Lambda}_q^{\perp}(\mathsf{Rot}(\mathbf{A}))$ and $\mathbf{\Lambda}_q(\mathsf{Rot}(\mathbf{A}^T)^T)$ over $\mathbb{Z}^{n\bar{m}}$ under the map $\phi : R \to \mathbb{Z}^n$, respectively. The $\mathbf{G}$-trapdoor notion can be easily extended to the ring setting. Let k be an integer and let $q = 3^k$. Let

$\mathbf{G}_3 = (1, 3, \ldots, 3^{k-1})^T \in R_q^{1\times k}$ be the public primitive vector, we have that $\mathsf{Rot}(\mathbf{G}_3) = (\mathbf{I}_n, 3\mathbf{I}_n, \ldots, 3^{k-1}\mathbf{I}_n) \in \mathbb{Z}_q^{n\times nk}$. We have the following results.

Proposition 2 ([47,26]). *Given any integers $n, w, k \geq 1$, $q = 3^k$, sufficiently large $\bar{m} = w + k$ and a tag $h \in R_q$, there is an efficient randomized algorithm $\mathsf{TrapGen}_R(1^n, 1^{\bar{m}}, q, h)$ that outputs a matrix $\mathbf{A} \in R_q^{1\times \bar{m}}$ and a $\mathbf{G}$-trapdoor $\mathbf{R} \in R_q^{w\times k}$ with quality $s_1(\mathbf{R}) \leq s\sqrt{n} \cdot O(\sqrt{w} + \sqrt{k} + \omega(\sqrt{\log n})) \cdot \omega(\sqrt{\log nw})$ such that the distribution of $\mathbf{a}$ is $\mathrm{negl}(n)$-far from uniform and $\mathbf{A}\begin{bmatrix}\mathbf{R}\\ \mathbf{I}_k\end{bmatrix} = h\mathbf{G}_3$.*

In addition, given a $\mathbf{G}$-trapdoor $\mathbf{R}$ of $\mathbf{A} \in R_q^{1\times\bar{m}}$ for some invertible tag $h \in R_q$, any $\mathbf{U} \in R_q^{1\times n'}$ for some integer $n' \geq 1$ and real $s \geq s_1(\mathbf{R}) \cdot \omega(\sqrt{\log n})$, there is an algorithm $\mathsf{SampleD}_R(\mathbf{R}, \mathbf{A}, h, \mathbf{U}, s)$ that samples from a distribution within $\mathrm{negl}(n)$ statistical distance of $\mathbf{E} \sim (D_{R,s})^{\bar{m}\times n'}$ satisfying $\mathbf{AE} = \mathbf{U}$.

Lemma 9 ([26]). *Let $n \geq 4$ be a power of 2, $q \geq 3$ a power of 3, $R_q = \mathbb{Z}_q[x]/(x^n + 1)$. Let integers $w \geq 2\lceil \log_2 q\rceil + 2$ and real $s \geq \omega(\sqrt{\log nw})$. With overwhelming probability over the choice of $\mathbf{A} \leftarrow R_q^{1\times w}$, if $\mathbf{r} \leftarrow (D_{R,s})^w$, then $\mathbf{Ar}$ is within negligible statistical distance from uniform distribution over R_q.*

2.4 LWE and (I)SIS Problems

In this paper, let ring $\mathcal{R}$ be either the integer ring $\mathbb{Z}$ or the polynomial ring $\mathbb{Z}[x]/(x^n + 1)$ with n being a power of 2. Let $\mathcal{R}_q = \mathcal{R}/(q\mathcal{R})$ be the quotient ring.

Learning with Errors (LWE) over ring $\mathcal{R}$. For any positive integer n, q, real $\alpha > 0$, and any vector $\mathbf{s} \in \mathcal{R}_q^n$, the distribution $A_{\mathbf{s},\alpha}$ over $\mathcal{R}_q^n \times \mathcal{R}_q$ is defined as $A_{\mathbf{s},\alpha} = \{(\mathbf{a}, \mathbf{a}^T\mathbf{s} + x) : \mathbf{a} \leftarrow \mathcal{R}_q^n, x \leftarrow D_{\mathcal{R},\alpha q}\}$, where $D_{\mathcal{R},\alpha q}$ is the discrete Gaussian distribution over $\mathcal{R}$ with parameter αq. For m independent samples $(\mathbf{a}_1, y_1), \ldots, (\mathbf{a}_m, y_m)$ from $A_{\mathbf{s},\alpha}$, we denote it in matrix form $(\mathbf{A}, \mathbf{y}) \in \mathcal{R}_q^{n\times m} \times \mathcal{R}_q^m$, where $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_m)$ and $\mathbf{y} = (y_1, \ldots, y_m)^T$. We say that an algorithm solves the $\mathrm{LWE}_{q,\alpha}$ problem over ring $\mathcal{R}$ if, for uniformly random $\mathbf{s} \leftarrow \mathcal{R}_q^n$, given polynomial samples from $A_{\mathbf{s},\alpha}$ it outputs $\mathbf{s}$ with noticeable probability. The decisional variant of LWE is that, for a uniformly random $\mathbf{s} \leftarrow \mathcal{R}_q^n$, the solving algorithm is asked to distinguish $A_{\mathbf{s},\alpha}$ from the uniform distribution over $\mathcal{R}_q^n \times \mathcal{R}_q$ (with only polynomial samples). For certain modulus q, the average-case decisional LWE problem is polynomially equivalent to its worst-case search version [51,46].

Note that the above LWE definition with $\mathcal{R} = \mathbb{Z}$ actually refers to the standard LWE problems in [51]. Moreover, the setting with $\mathcal{R} = R$ refers to the standard ring-LWE problems [46].

Small Integer Solutions (SIS) over ring $\mathcal{R}$. The Small Integer Solution (SIS) problem was first introduced by Ajtai [4]. Formally, given positive $n, m, q \in \mathbb{Z}$, a real $\beta > 0$, and a uniformly random matrix $\mathbf{A} \in \mathcal{R}_q^{n\times m}$, the $\mathrm{SIS}_{q,m,\beta}$ problem over $\mathcal{R}$ asks to find a non-zero vector $\mathbf{e} \in \mathcal{R}^m$ such that $\mathbf{Ae} = \mathbf{0} \mod q$ and $\|\mathbf{e}\| \leq \beta$. In [31], Gentry et al. introduced the ISIS problem, which was an inhomogeneous variant of SIS. Specifically, given an extra random syndrome $\mathbf{u} \in \mathcal{R}_q^n$, the $\mathrm{ISIS}_{q,m,\beta}$ problem over $\mathcal{R}$ asks to find a vector $\mathbf{e} \in \mathcal{R}^m$ such that $\mathbf{A}\mathbf{e} = \mathbf{u}$ and $\|\mathbf{e}\| \leq \beta$.

2.5 Digital Signatures

A digital signature scheme $\mathcal{SIG} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ consists of three PPT algorithms. Taking the security parameter κ as input, the key generation algorithm outputs a verification key vk and a secret signing key sk, i.e., $(vk, sk) \leftarrow \mathsf{KeyGen}(1^\kappa)$. The signing algorithm takes vk, sk and a message $M \in \{0,1\}^*$ as inputs, outputs a signature σ on M, briefly denoted as $\sigma \leftarrow \mathsf{Sign}(sk, M)$. The verification algorithm takes vk, message $M \in \{0,1\}^*$ and a string $\sigma \in \{0,1\}^*$ as inputs, outputs 1 if σ is a valid signature on M, else outputs 0, denoted as $1/0 \leftarrow \mathsf{Verify}(vk, M, \sigma)$. For correctness, we require that for any $(vk, sk) \leftarrow \mathsf{KeyGen}(1^\kappa)$, any message $M \in \{0,1\}^*$, and any $\sigma \leftarrow \mathsf{Sign}(sk, M)$, the equation $\mathsf{Verify}(vk, M, \sigma) = 1$ holds with overwhelming probability, where the probability is taken over the choices of the random coins used in KeyGen, Sign and Verify.

The standard security notion for digital signature scheme is the existential unforgeability against chosen message attacks (EUF-CMA), which (informally) says that any PPT forger, after seeing valid signatures on a polynomial number of adaptively chosen messages, cannot create a valid signature on a new message. Formally, consider the following game between a challenger $\mathcal{C}$ and a forger $\mathcal{F}$:

KeyGen. The challenger $\mathcal{C}$ first runs $(vk, sk) \leftarrow \mathsf{KeyGen}(1^\kappa)$ with the security parameter κ. Then, it gives the verification key vk to the forger $\mathcal{F}$, and keeps the signing secret key sk to itself.

Signing. The forger $\mathcal{F}$ is allowed to ask the signature on any message M. The challenger $\mathcal{C}$ computes and sends $\sigma \leftarrow \mathsf{Sign}(sk, M)$ to the forger $\mathcal{F}$. The forger can repeat this any polynomial number of times.

Forge. $\mathcal{F}$ outputs a message-signature pair (M^*, σ^*). Let $\mathcal{Q}$ be the set of all messages queried by $\mathcal{F}$ in the signing phase. If $M^* \notin \mathcal{Q}$ and $\mathsf{Verify}(vk, M^*, \sigma^*) = 1$, the challenger $\mathcal{C}$ outputs 1, else outputs 0.

We say that $\mathcal{F}$ wins the game if the challenger $\mathcal{C}$ outputs 1. The advantage of $\mathcal{F}$ in the above security game is defined as $\mathrm{Adv}_{\mathcal{SIG},\mathcal{F}}^{\text{euf-cma}}(1^\kappa) = \Pr[\mathcal{C} \text{ outputs } 1]$.

Definition 2 (EUF-CMA Security). *Let κ be the security parameter. A signature scheme $\mathcal{SIG}$ is said to be existentially unforgeable against chosen message attacks (*EUF-CMA*) if the advantage $\mathrm{Adv}_{\mathcal{SIG},\mathcal{F}}^{\text{euf-cma}}(1^\kappa)$ is negligible in κ for any PPT forger $\mathcal{F}$.*

In a modified security game of existential unforgeability against non-adaptive chosen message attacks, $\mathcal{F}$ is asked to output all the messages $\mathcal{Q} = \{M_1, \ldots, M_Q\}$ for signing queries before seeing the verification key vk, and is given vk and the signatures $\{\sigma_1, \ldots, \sigma_Q\}$ on all the queried messages at the same time (i.e., there is no adaptive signing query phase). The resulting security notion defined using the modified game as in Definition 2 is denoted as EUF-naCMA. One can transform an EUF-naCMA secure signature scheme into an EUF-CMA secure one [9,26] by using chameleon hash functions [42]. Besides, if the number of the signing queries allowed for the adversary is upper bounded by some (arbitrary) polynomial that needs to be fixed before the key generation phase in the security game, the resulting security notion is denoted as EUF-qCMA.

2.6 Identity-based Encryption

An identity-based encryption (IBE) scheme consists of four PPT algorithms $\mathcal{IBE} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$. Taking the security parameter κ as input, the randomized key generation algorithm Setup outputs a master public key mpk and a master secret key msk, denoted as $(mpk, msk) \leftarrow \mathsf{Setup}(1^\kappa)$. The (randomized) extract algorithm takes mpk, msk and an identity id as inputs, outputs a user private key sk_{id} for id, briefly denoted as $sk_{id} \leftarrow \mathsf{Extract}(msk, id)$. The randomized encryption algorithm Enc takes mpk, id and a plaintext M as inputs, outputs a ciphertext C, denoted as $C \leftarrow \mathsf{Enc}(mpk, id, M)$. The deterministic algorithm Dec takes sk_{id} and C as inputs, outputs a plaintext M, or a special symbol $\perp$, which is denoted as $M/\perp \leftarrow \mathsf{Dec}(sk_{id}, C)$. In addition, for all $(mpk, msk) \leftarrow \mathsf{Setup}(1^\kappa), sk_{id} \leftarrow \mathsf{Extract}(msk, id)$ and any plaintext M, we require that $\mathsf{Dec}(sk_{id}, C) = M$ holds for any $C \leftarrow \mathsf{Enc}(mpk, id, M)$ with overwhelming probability.

The standard semantic security of IBE was first introduced in [11]. Formally, consider the following game played by an adversary $\mathcal{A}$.

Setup. The challenger $\mathcal{C}$ first runs $\mathsf{Setup}(1^\kappa)$ with the security parameter κ. Then, it gives the adversary $\mathcal{A}$ the master public key mpk, and keeps the master secret key msk to itself.

Phase 1. The adversary is allowed to query the user private key for any identity id. The challenger $\mathcal{C}$ runs $sk_{id} \leftarrow \mathsf{Extract}(msk, id)$ and sends sk_{id} to the adversary $\mathcal{A}$. The adversary can repeat the user private key query any polynomial number of times.

Challenge. The adversary $\mathcal{A}$ outputs a pair of challenge plaintext (M_0, M_1) and a challenge identity id^* with a restriction that id^* is not used in the user private key query in phase 1. The challenger $\mathcal{C}$ first chooses a uniformly random $b^* \leftarrow \{0, 1\}$, and then computes $C_{b^*} \leftarrow \mathsf{Enc}(mpk, id^*, M_{b^*})$. Finally, it sends C_{b^*} as the challenge ciphertext to $\mathcal{A}$.

Phase 2. The adversary can adaptively make more user private key queries with any identity $id \neq id^*$. The challenger $\mathcal{C}$ responds as in Phase 1.

Guess. Finally, $\mathcal{A}$ outputs a guess $b \in \{0, 1\}$. If $b = b^*$, the challenger $\mathcal{C}$ outputs 1, else outputs 0.

The advantage of $\mathcal{A}$ in the above security game is defined as $\mathrm{Adv}^{\text{ind-id-cpa}}_{\mathcal{IBE},\mathcal{A}}(\kappa) = |\Pr[b = b^*] - \frac{1}{2}|$.

Definition 3 (IND-ID-CPA Security). *We say an IBE scheme $\mathcal{IBE}$ is* IND-ID-CPA *secure if for any PPT adversary $\mathcal{A}$, its advantage* $\mathrm{Adv}^{\text{ind-id-cpa}}_{\mathcal{IBE},\mathcal{A}}(\kappa)$ *is negligible in κ.*

In the security game against chosen ciphertext attacks (i.e., IND-ID-CCA), the adversary is also allowed to make decryption queries in both Phase 1 and Phase 2 such that it can obtain the decrypted results from any identity-ciphertext pair $(id, C) \neq (id^*, C_{b^*})$. Besides, the paper [15] also introduced a weaker security notion, known as selective-identity security, by using a modified security game, where the adversary is asked to output the challenge identity id^* before seeing the master public key in the setup phase, and is restricted to make user private key query for $id \neq id^*$ in both Phase 1 and Phase 2. The resulting security notion defined using the modified game as in Definition 3 is denoted as IND-sID-CPA. Furthermore, if the number of the private key queries allowed for the adversary is upper bounded by some (arbitrary) polynomial that needs to be fixed before the key generation phase in the security game, the resulting security notion is denoted as IND-qID-CPA.

3 Programmable Hash Functions from Lattices

Let $\ell, m, \bar{m}, n, \bar{n}, q, u, v \in \mathbb{Z}$ be some polynomials in the security parameter κ. Let ring $\mathcal{R}$ be either the integer ring $\mathbb{Z}$ or the polynomial ring $\mathbb{Z}[x]/(x^n+1)$. Let $\mathcal{R}_q = \mathcal{R}/(q\mathcal{R})$ be the quotient ring. We now give the definition of lattice-based programmable hash functions (PHF). By $\mathcal{I}_{\bar{n}}$ we denote the set of invertible matrices in $\mathcal{R}_q^{\bar{n}\times\bar{n}}$. A hash function $\mathcal{H} : \mathcal{X} \to \mathcal{R}_q^{\bar{n}\times m}$ defined over the ring $\mathcal{R}$ consists of two algorithms $(\mathcal{H}.\mathrm{Gen}_{\mathcal{R}}, \mathcal{H}.\mathrm{Eval}_{\mathcal{R}})$. Given the security parameter κ, the probabilistic polynomial time (PPT) key generation algorithm $\mathcal{H}.\mathrm{Gen}_{\mathcal{R}}(1^\kappa)$ outputs a key K, i.e., $K \leftarrow \mathcal{H}.\mathrm{Gen}_{\mathcal{R}}(1^\kappa)$. For any input $X \in \mathcal{X}$, the efficiently deterministic evaluation algorithm $\mathcal{H}.\mathrm{Eval}_{\mathcal{R}}(K, X)$ outputs a hash value $\mathbf{Z} \in \mathcal{R}_q^{\bar{n}\times m}$, i.e., $\mathbf{Z} = \mathcal{H}.\mathrm{Eval}_{\mathcal{R}}(K, X)$. For simplicity, we write $\mathrm{H}_K(X) = \mathcal{H}.\mathrm{Eval}_{\mathcal{R}}(K, X)$.

Definition 4 (Lattice-based Programmable Hash Functions). *A hash function $\mathcal{H} : \mathcal{X} \to \mathcal{R}_q^{\bar{n}\times m}$ defined over ring $\mathcal{R}$ is a $(u, v, \beta, \gamma, \delta)$-*PHF *if there exist a PPT trapdoor key generation algorithm $\mathcal{H}.\mathrm{TrapGen}_{\mathcal{R}}$, and an efficiently deterministic trapdoor evaluation algorithm $\mathcal{H}.\mathrm{TrapEval}_{\mathcal{R}}$ such that for a uniformly random matrix $\mathbf{A} \leftarrow \mathcal{R}_q^{\bar{n}\times\bar{m}}$ and a (public) trapdoor matrix $\mathbf{B} \in \mathcal{R}_q^{\bar{n}\times m}$,*[7] *the following properties hold:*

Syntax: *The PPT algorithm $(K', td) \leftarrow \mathcal{H}.\mathrm{TrapGen}_{\mathcal{R}}(1^\kappa, \mathbf{A}, \mathbf{B})$ outputs a key K' together with a trapdoor td. Moreover, for any input $X \in \mathcal{X}$, the deterministic algorithm $(\mathbf{R}_X, \mathbf{S}_X) = \mathcal{H}.\mathrm{TrapEval}_{\mathcal{R}}(td, K', X)$ returns $\mathbf{R}_X \in \mathcal{R}_q^{\bar{m}\times m}$ and $\mathbf{S}_X \in \mathcal{R}_q^{\bar{n}\times\bar{n}}$ such that $s_1(\mathbf{R}_X) \leq \beta$ and $\mathbf{S}_X \in \mathcal{I}_{\bar{n}} \cup \{\mathbf{0}\}$ hold with overwhelming probability over the trapdoor td that is produced along with K'.*

Correctness: *For all possible $(K', td) \leftarrow \mathcal{H}.\mathrm{TrapGen}_{\mathcal{R}}(1^\kappa, \mathbf{A}, \mathbf{B})$, all $X \in \mathcal{X}$ and its corresponding $(\mathbf{R}_X, \mathbf{S}_X) = \mathcal{H}.\mathrm{TrapEval}_{\mathcal{R}}(td, K', X)$, we have $\mathrm{H}_{K'}(X) = \mathcal{H}.\mathrm{Eval}_{\mathcal{R}}(K', X) = \mathbf{A}\mathbf{R}_X + \mathbf{S}_X\mathbf{B}$.*

[7] A general trapdoor matrix $\mathbf{B}$ is used for utmost generality, but the publicly known trapdoor matrix $\mathbf{B} = \mathbf{G}$ in [47] is recommended for both efficiency and simplicity.

Statistically close trapdoor keys: *For all* $(K', td) \leftarrow \mathcal{H}.\mathrm{TrapGen}_{\mathcal{R}}(1^\kappa, \mathbf{A}, \mathbf{B})$ *and* $K \leftarrow \mathcal{H}.\mathrm{Gen}_{\mathcal{R}}(1^\kappa)$*, the statistical distance between* $(\mathbf{A}, K')$ *and* $(\mathbf{A}, K)$ *is at most* γ*.*

Well-distributed hidden matrices: *For all* $(K', td) \leftarrow \mathcal{H}.\mathrm{TrapGen}_{\mathcal{R}}(1^\kappa, \mathbf{A}, \mathbf{B})$*, any inputs* $X_1, \ldots, X_u, Y_1, \ldots, Y_v \in \mathcal{X}$ *such that* $X_i \neq Y_j$ *for any* i, j*, let* $(\mathbf{R}_{X_i}, \mathbf{S}_{X_i}) = \mathcal{H}.\mathrm{TrapEval}(td, K', X_i)$ *and* $(\mathbf{R}_{Y_i}, \mathbf{S}_{Y_i}) = \mathcal{H}.\mathrm{TrapEval}_{\mathcal{R}}(td, K', Y_i)$*. Then, we have that*

$$\Pr[\mathbf{S}_{X_1} = \cdots = \mathbf{S}_{X_u} = \mathbf{0} \wedge \mathbf{S}_{Y_1}, \ldots, \mathbf{S}_{Y_v} \in \mathcal{I}_{\bar{n}}] \geq \delta,$$

where the probability is over the trapdoor td produced along with K'*.*

If γ *is negligible and* $\delta > 0$ *is noticeable, we simply say that* $\mathcal{H}$ *is a* (u, v, β)-PHF*. Furthermore, if* $\mathcal{H}$ *is a* (u, v, β)-PHF *for every polynomial u (resp. v) in* κ*, we say that* $\mathcal{H}$ *is a* $(\mathsf{poly}, v, \beta)$-PHF *(resp.* $(u, \mathsf{poly}, \beta)$-PHF*).*

A ***weak programmable hash function*** is a relaxed version of PHF, where the $\mathcal{H}$.TrapGen algorithm additionally takes a list $X_1, \ldots, X_u \in \mathcal{X}$ as inputs such that the well-distributed hidden matrices property holds in the following sense: For all $(K', td) \leftarrow \mathcal{H}.\mathrm{TrapGen}(1^\kappa, \mathbf{A}, \mathbf{B}, \{X_1, \ldots, X_u\})$, any inputs $Y_1, \ldots, Y_v \in \mathcal{X}$ such that $Y_j \notin \{X_1, \ldots, X_u\}$ for all j, let $(\mathbf{R}_{X_i}, \mathbf{S}_{X_i}) = \mathcal{H}.\mathrm{TrapEval}(td, K', X_i)$ and $(\mathbf{R}_{Y_i}, \mathbf{S}_{Y_i}) = \mathcal{H}.\mathrm{TrapEval}(td, K', Y_i)$, we have that $\Pr[\mathbf{S}_{X_1} = \cdots = \mathbf{S}_{X_u} = \mathbf{0} \wedge \mathbf{S}_{Y_1}, \ldots, \mathbf{S}_{Y_v} \in \mathcal{I}_n] \geq \delta$, where the probability is over the trapdoor td produced along with K'.

Besides, a hash function $\mathcal{H} : \mathcal{X} \rightarrow \mathcal{R}_q^{\bar{n} \times m}$ can be a (weak) (u, v, β)-PHF for different parameters u and v, since there might exist different pairs of trapdoor key generation and trapdoor evaluation algorithms for $\mathcal{H}$. If this is the case, one can easily show that the keys output by these trapdoor key generation algorithms are statistically indistinguishable by definition.

3.1 Type-I Construction

We describe the Type-I construction of lattice-based PHFs in the following.

Definition 5. *Let* $\ell, \bar{n}, m, q \in \mathbb{Z}$ *be some polynomials in the security parameter* κ*. Let* E *be a deterministic encoding from* $\mathcal{X}$ *to* $(\mathcal{R}_q^{\bar{n} \times \bar{n}})^\ell$*, the hash function* $\mathcal{H} = (\mathcal{H}.\mathrm{Gen}, \mathcal{H}.\mathrm{Eval})$ *with key space* $\mathcal{K} \subseteq (\mathcal{R}_q^{\bar{n} \times m})^{\ell+1}$ *is defined as follows:*

- $\mathcal{H}.\mathrm{Gen}(1^\kappa)$*: Randomly choose* $(\mathbf{A}_0, \ldots, \mathbf{A}_\ell) \leftarrow \mathcal{K}$*, return* $K = \{\mathbf{A}_i\}_{i \in \{0, \ldots, \ell\}}$*.*
- $\mathcal{H}.\mathrm{Eval}(K, X)$*: Let* $\mathrm{E}(X) = (\mathbf{C}_1, \ldots, \mathbf{C}_\ell)$*, return* $\mathbf{Z} = \mathbf{A}_0 + \sum_{i=1}^{\ell} \mathbf{C}_i \mathbf{A}_i$*.*

We note that the above hash function has actually been (implicitly) used to construct both signatures (e.g, [13,9,49]) and encryptions (e.g., [2,47]). Let $\mathbf{I}_{\bar{n}}$ be the $\bar{n} \times \bar{n}$ identity matrix in $\mathcal{R}^{\bar{n} \times \bar{n}}$. In the following theorems, we summarize several known results which were implicitly proved in [13,2,47].

Theorem 1. *Let $\mathcal{K} = (\mathbb{Z}_q^{\bar{n}\times m})^{\ell+1}$ and $\mathcal{X} = \{0,1\}^\ell$. In addition, given an input $X = (X_1, \ldots, X_\ell) \in \mathcal{X}$, the encoding function $\mathrm{E}(X)$ returns $\mathbf{C}_i = (-1)^{X_i} \cdot \mathbf{I}_{\bar{n}}$ for $i = \{1, \ldots, \ell\}$. Then, for large enough integer $m = O(n \log q)$, the instantiated hash function $\mathcal{H}$ over ring $\mathcal{R} = \mathbb{Z}$ of Definition 5 is a $(1, \mathsf{poly}, \beta)$-PHF for some $\beta \leq \sqrt{\ell m} \cdot \omega(\sqrt{\log n})$.*

Theorem 2. *For large enough $m = O(\bar{n} \log q)$, the hash function $\mathcal{H}$ over $\mathcal{R} = \mathbb{Z}$ given in Definition 5 is a weak $(1, \mathsf{poly}, \beta, \gamma, \delta)$-PHF with $\beta \leq \sqrt{\ell m} \cdot \omega(\sqrt{\log \bar{n}})$, $\gamma = \mathrm{negl}(\kappa)$, and $\delta = 1$ when instantiated as follows:*

- *Let $\mathcal{K} = (\mathbb{Z}_q^{\bar{n}\times m})^2$ (i.e., $\ell = 1$) and $\mathcal{X} = \mathbb{Z}_q^{\bar{n}}$. Given an input $X \in \mathcal{X}$, the encoding $\mathrm{E}(X)$ returns $H(X)$ where $H : \mathbb{Z}_q^{\bar{n}} \to \mathbb{Z}_q^{\bar{n}\times\bar{n}}$ is an FRD encoding.*
- *Let $\mathcal{K} = (\mathbb{Z}_q^{\bar{n}\times m})^{\ell+1}$ and $\mathcal{X} = \{0,1\}^\ell$. Given an input $X = (X_1, \ldots, X_\ell) \in \mathcal{X}$, the encoding $\mathrm{E}(X)$ returns $\mathbf{C}_i = X_i \cdot \mathbf{I}_{\bar{n}}$ for all $i \in \{1, \ldots, \ell\}$.*

We first note that the above two theorems can be easily extended to $\mathcal{R} = R$. Besides, unlike the traditional PHFs [35,34,17] where a bigger u is usually better in constructing short signature schemes, our lattice-based PHFs seem more useful when the parameter v is bigger (e.g., a polynomial in κ). There is a simple explanation: although both notions aim at capturing some kind of partitioning proof trick, i.e., each programmed hash value contains a hidden element behaving as a trigger of some prior embedded trapdoors, for traditional PHFs the trapdoor is usually triggered when the hidden element is zero, while in the lattice setting the trapdoor is typically triggered when the hidden element is a non-zero invertible one. This also explains why previous known constructions on lattices (e.g., the instantiations in Theorem 1 and Theorem 2) are (weak) $(1, \mathsf{poly}, \beta)$-PHFs for any polynomial $v = \mathsf{poly}(\kappa) \in \mathbb{Z}$ and real $\beta \in \mathbb{R}$.

3.2 Type-II Construction

Let integers $\ell, \bar{m}, n, \bar{n}, q, u, v, L, N$ be some polynomials in the security parameter κ, and let $k = \lceil \log_2 q \rceil$. We now exploit the nice property of the publicly known trapdoor matrix $\mathbf{B} = \mathbf{G} \in \mathcal{R}_q^{\bar{n}\times\bar{n}k}$ to construct more efficient PHF from lattices for an arbitrary but fixed polynomial $v = \mathsf{poly}(\kappa)$.

Definition 6. *Let $n, q \in \mathbb{Z}$ be some polynomials in the security parameter κ. For any $\ell, v \in \mathbb{Z}$ and $L = 2^\ell$, let $N \leq 16v^2\ell, \eta \leq 4v\ell$ and $CF = \{CF_X\}_{X\in[L]}$ be defined as in Lemma 1. Let $\mu = \lceil \log_2 N \rceil$ and $k = \lceil \log_2 q \rceil$. Then, the hash function $\mathcal{H} = (\mathcal{H}.\mathrm{Gen}_{\mathcal{R}}, \mathcal{H}.\mathrm{Eval}_{\mathcal{R}})$ from $[L]$ to $\mathcal{R}_q^{\bar{n}\times\bar{n}k}$ is defined as follows:*

- *$\mathcal{H}.\mathrm{Gen}(1^\kappa)$: Randomly choose $\hat{\mathbf{A}}, \mathbf{A}_i \leftarrow \mathcal{R}_q^{\bar{n}\times\bar{n}k}$ for $i \in \{0, \ldots, \mu-1\}$, return the key $K = (\hat{\mathbf{A}}, \{\mathbf{A}_i\}_{i\in\{0,\ldots,\mu-1\}})$.*
- *$\mathcal{H}.\mathrm{Eval}(K, X)$: Given the key $K = (\hat{\mathbf{A}}, \{\mathbf{A}_i\}_{i\in\{0,\ldots,\mu-1\}})$ and an integer $X \in [L]$ as inputs, first compute $\mathbf{B}_z = \mathsf{Comp}_N(\{\mathbf{A}_i\}_{0\leq i\leq\mu-1}, z)$ for all $z \in CF_X$ as shown in* Fig. 1. *Then, return $\mathbf{Z} = \hat{\mathbf{A}} + \sum_{z\in CF_X} \mathbf{B}_z$.*

```
$\mathsf{Comp}_N(\{\mathbf{A}_i\}_{0\le i\le \mu-1}, z)$
$(b_0,\ldots,b_{\mu-1}) := \mathsf{BitDecomp}_N(z)$
$\mathbf{B}_z := \mathbf{A}_{\mu-1} - b_{\mu-1}\cdot\mathbf{G}$
For $i = \mu-2,\ldots,0$
    $\mathbf{B}_z := (\mathbf{A}_i - b_i\cdot\mathbf{G})\cdot\mathbf{G}^{-1}(\mathbf{B}_z)$
Return $\mathbf{B}_z$
```

```
$\mathsf{Tcomp}_N(\{\mathbf{A}_i,\mathbf{R}_i\}_{0\le i\le \mu-1}, z)$
$(b_0,\ldots,b_{\mu-1}) := \mathsf{BitDecomp}_N(z)$
$\mathbf{B}_z := \mathbf{A}_{\mu-1} - b_{\mu-1}\cdot\mathbf{G}$
$\mathbf{R}_z := \mathbf{R}_{\mu-1}$
$\mathbf{S}_z := (1 - b^*_{\mu-1} - b_{\mu-1})\cdot\mathbf{I}_n$
For $i = \mu-2,\ldots,0$
    $\mathbf{B}_z := (\mathbf{A}_i - b_i\cdot\mathbf{G})\cdot\mathbf{G}^{-1}(\mathbf{B}_z)$
    $\mathbf{R}_z := \mathbf{R}_i\cdot\mathbf{G}^{-1}(\mathbf{B}_z) + (1 - b^*_i - b_i)\cdot\mathbf{R}_z$
    $\mathbf{S}_z := (1 - b^*_i - b_i)\cdot\mathbf{S}_z$
Return $(\mathbf{R}_z,\mathbf{S}_z)$
```

Fig. 1. The Algorithms Used in Definition 6 and Theorem 3

In the following, we now show that for any prior fixed $v = \mathsf{poly}(\kappa)$, the hash function $\mathcal{H}$ with $\mathcal{R} = \mathbb{Z}$ given in Definition 6 is a $(1, v, \beta)$-PHF for some polynomially bounded $\beta \in \mathbb{R}$. We also note that similar results can also be obtained for $\mathcal{R} = R$.

Theorem 3. *For any $\ell, v \in \mathbb{Z}$ and $L = 2^\ell$, let $N \le 16v^2\ell, \eta \le 4v\ell$ and $CF = \{CF_X\}_{X\in[L]}$ be defined as in Lemma 1. Then, for large enough $\bar{m} = O(\bar{n}\log q)$, the hash function $\mathcal{H}$ with $\mathcal{R} = \mathbb{Z}$ in Definition 6 is a $(1, v, \beta, \gamma, \delta)$-PHF with $\beta \le \mu v\ell\bar{m}^{1.5}\cdot\omega(\sqrt{\log\bar{m}})$, $\gamma = \mathrm{negl}(\kappa)$ and $\delta = 1/N - \mathrm{negl}(\kappa)$, where $\mu = \lceil\log_2 N\rceil$.*

In particular, if we set $\ell = \bar{n}$ and $v = \omega(\log\bar{n})$, then $\beta = \tilde{O}(\bar{n}^{2.5})$, and the key of $\mathcal{H}$ only consists of $\mu = O(\log\bar{n})$ matrices.

Proof. We now construct a pair of trapdoor algorithms for $\mathcal{H}$ as follows:

- $\mathcal{H}.\mathrm{TrapGen}_{\mathbb{Z}}(1^\kappa, \mathbf{A}, \mathbf{G})$: Given a uniformly random $\mathbf{A} \leftarrow \mathbb{Z}_q^{\bar{n}\times\bar{m}}$ and a matrix $\mathbf{G} \in \mathbb{Z}_q^{\bar{n}\times\bar{n}k}$ for sufficiently large $\bar{m} = O(\bar{n}\log q)$ as inputs, let $s = \omega(\sqrt{\log\bar{m}}) \in \mathbb{R}$ satisfy the requirement in Lemma 5. Randomly choose $\hat{\mathbf{R}}, \mathbf{R}_i \leftarrow (D_{\mathbb{Z}^{\bar{m}},s})^{\bar{n}k}$ for $i \in \{0,\ldots,\mu-1\}$, and an integer $z^* \leftarrow [N]$. Let $(b^*_0,\ldots,b^*_{\mu-1}) = \mathsf{BitDecomp}_N(z^*)$, and let c be the number of 1's in the vector $(b^*_0,\ldots,b^*_{\mu-1})$. Then, compute $\hat{\mathbf{A}} = \mathbf{A}\hat{\mathbf{R}} - (-1)^c\cdot\mathbf{G}$ and $\mathbf{A}_i = \mathbf{A}\mathbf{R}_i + (1 - b^*_i)\cdot\mathbf{G}$. Finally, return the key $K' = (\hat{\mathbf{A}}, \{\mathbf{A}_i\}_{i\in\{0,\ldots,\mu-1\}})$ and the trapdoor $td = (\hat{\mathbf{R}}, \{\mathbf{R}_i\}_{i\in\{0,\ldots,\mu-1\}}, z^*)$.
- $\mathcal{H}.\mathrm{TrapEval}_{\mathbb{Z}}(td, K', X)$: Given the trapdoor $td = (\hat{\mathbf{R}}, \{\mathbf{R}_i\}_{i\in\{0,\ldots,\mu-1\}}, z^*)$ for the key $K' = (\hat{\mathbf{A}}, \{\mathbf{A}_i\}_{i\in\{0,\ldots,\mu-1\}})$ and an input $X \in [L]$ as inputs, the algorithm first computes CF_X by Lemma 1. Then, let $(b^*_0,\ldots,b^*_{\mu-1}) = \mathsf{BitDecomp}_N(z^*)$, and compute $(\mathbf{R}_z, \mathbf{S}_z) = \mathsf{Tcomp}_N(\{\mathbf{A}_i,\mathbf{R}_i\}_{0\le i\le\mu-1}, z)$ for all $z \in CF_X$ as shown in Fig. 1. Finally, return $\mathbf{R}_X = \hat{\mathbf{R}} + \sum_{z\in CF_X}\mathbf{R}_z$ and $\mathbf{S}_X = -(-1)^c\cdot\mathbf{I}_n + \sum_{z\in CF_X}\mathbf{S}_z$.

Since $s \ge \omega(\sqrt{\log\bar{m}})$ and $\hat{\mathbf{R}}, \mathbf{R}_i \leftarrow (D_{\mathbb{Z}^{\bar{m}},s})^{\bar{n}k}$, each matrix in the key $K' = (\hat{\mathbf{A}}, \{\mathbf{A}_i\}_{i\in\{0,\ldots,\mu-1\}})$ is statistically close to uniform over $\mathbb{Z}_q^{\bar{n}\times\bar{n}k}$ by Lemma 5.

Using a standard hybrid argument, it is easy to show that the statistical distance γ between $(\mathbf{A}, K')$ and $(\mathbf{A}, K)$ is negligible, where $K \leftarrow \mathcal{H}.\mathrm{Gen}(1^\kappa)$. In particular, this means that z^* is statistically hidden in K'.

For correctness, we first show that $\mathbf{B}_z = \mathbf{A}\mathbf{R}_z + \mathbf{S}_z\mathbf{G}$ always holds during the computation of Tcomp. By definition, we have that $\mathbf{B}_z = \mathbf{A}_{\mu-1} - b_{\mu-1} \cdot \mathbf{G} = \mathbf{A}\mathbf{R}_z + \mathbf{S}_z\mathbf{G}$ holds before entering the loop. Assume that $\mathbf{B}_z = \mathbf{A}\mathbf{R}_z + \mathbf{S}_z\mathbf{G}$ holds before entering the j-th (i.e., $i = j$) iteration of the loop, we now show that the equation $\mathbf{B}_z = \mathbf{A}\mathbf{R}_z + \mathbf{S}_z\mathbf{G}$ still holds after the j-th iteration. Since $\mathbf{A}_j - b_j \cdot \mathbf{G} = \mathbf{A}\mathbf{R}_j + (1 - b_j^* - b_j) \cdot \mathbf{G}$, we have that $\mathbf{B}_z := (\mathbf{A}_j - b_j \cdot \mathbf{G}) \cdot \mathbf{G}^{-1}(\mathbf{B}_z) = \mathbf{A}\mathbf{R}_j \cdot \mathbf{G}^{-1}(\mathbf{B}_z) + (1 - b_j^* - b_j) \cdot (\mathbf{A}\mathbf{R}_z + \mathbf{S}_z\mathbf{G})$. This means that if we set $\mathbf{R}_z := \mathbf{R}_j \cdot \mathbf{G}^{-1}(\mathbf{B}_z) + (1 - b_j^* - b_j) \cdot \mathbf{R}_z$ and $\mathbf{S}_z := (1 - b_j^* - b_j) \cdot \mathbf{S}_z$, the equation $\mathbf{B}_z = \mathbf{A}\mathbf{R}_z + \mathbf{S}_z\mathbf{G}$ still holds. In particular, we have that $\mathbf{S}_z = \prod_{i=0}^{\mu-1}(1 - b_i^* - b_i) \cdot \mathbf{I}_{\bar{n}}$ holds at the end of the inner loop. It is easy to check that $\mathbf{S}_z = \mathbf{0}$ for any $z \neq z^*$, and $\mathbf{S}_z = (-1)^c \cdot \mathbf{I}_{\bar{n}}$ for $z = z^*$, where c is the number of 1's in the binary vector $(b_0^*, \dots, b_{\mu-1}^*) = \mathsf{BitDecomp}_N(z^*)$. The correctness of the trapdoor evaluation algorithm follows from that fact that $\mathbf{Z} = \mathcal{H}.\mathrm{Eval}(K', X) = \hat{\mathbf{A}} + \sum_{z \in CF_X} \mathbf{B}_z = \mathbf{A}\hat{\mathbf{R}} - (-1)^c \cdot \mathbf{G} + \sum_{z \in CF_X}(\mathbf{A}\mathbf{R}_z + \mathbf{S}_z\mathbf{G}) = \mathbf{A}\mathbf{R}_X + \mathbf{S}_X\mathbf{B}$. In particular, we have that $\mathbf{S}_X = -(-1)^c \cdot \mathbf{I}_{\bar{n}}$ if $z^* \notin CF_X$, else $\mathbf{S}_X = \mathbf{0}$.

Since $s_1(\mathbf{G}^{-1}(\mathbf{B}_z)) \le \bar{n}k$ by the fact that $\mathbf{G}^{-1}(\mathbf{B}_z) \in \{0,1\}^{\bar{n}k \times \bar{n}k}$, and $s_1(\hat{\mathbf{R}}), s_1(\mathbf{R}_i) \le (\sqrt{\bar{m}} + \sqrt{\bar{n}k}) \cdot \omega(\sqrt{\log \bar{m}})$ by Lemma 3, we have that $s_1(\mathbf{R}_z) \le \mu\bar{m}^{1.5} \cdot \omega(\sqrt{\log \bar{m}})$ holds except with negligible probability for any $z \in CF_X$. Using $|CF_X| = \eta \le 4v\ell$, the inequality $s_1(\mathbf{R}_X) \le \mu v \ell \bar{m}^{1.5} \cdot \omega(\sqrt{\log \bar{m}})$ holds except with negligible probability for any $X \in [L]$. Besides, for any $X_1, Y_1, \dots, Y_v \in [L]$ such that $X_1 \neq Y_j$ for all $j \in \{1, \dots, v\}$, there is at least one element in $CF_{X_1} \subseteq [N]$ that does not belong to the union set $\cup_{j \in \{1,\dots,v\}} CF_{Y_j}$. This is because the family $CF = \{CF_X\}_{X \in [L]}$ is v-cover-free. Since z^* is randomly chosen from $[N]$ and is statistically hidden in the key K', the probability $\Pr[z^* \in CF_{X_1} \wedge z^* \notin \cup_{j \in \{1,\dots,v\}} CF_{Y_j}]$ is at least $1/N - \mathrm{negl}(\kappa)$. Thus, we have that $\Pr[\mathbf{S}_{X_1} = \mathbf{0} \wedge \mathbf{S}_{Y_1} = \cdots = \mathbf{S}_{Y_v} = -(-1)^c \cdot \mathbf{I}_n \in \mathcal{I}_n] \ge 1/N - \mathrm{negl}(\kappa)$. □

3.3 Type-III Construction

In this section, we present a PHF construction with constant keys over rings, which is inspired by the homomorphic equality testing algorithm [1]. At a high level, our construction heavily relies on the function $(2n)^{-1}\sum_{i=0}^{2n-1} x^i$ in $R_q = \mathbb{Z}[x]/(x^n + 1)$, where n is a power of 2 and q is an odd integer. In particular, given any $z^*, z \in [N]$ for some $N \le 2n$, we have that

$$f_{z^*,2n}(z) = (2n)^{-1}\sum_{i=0}^{2n-1}(x^{z^*-z})^i = \begin{cases} 1, & \text{if } z^* = z; \\ 0, & \text{otherwise.} \end{cases}$$

By using the fact that $f_{z^*,2n}(z) = \sum_{i=0}^{2n-1}\left((2n)^{-1}x^{-zi}\right)x^{z^*i} = \sum_{i=0}^{2n-1} c(z, 2n, i)u^i$ for some $c(z, 2n, i) = (2n)^{-1}x^{-zi}, u = x^{z^*}$, the authors [1] proposed a homomorphic equality testing algorithm which, given an encoding of $u = x^{z^*}$ for some unknown $z^* \in [N]$ and a public integer $z \in [N]$, outputs an encoding of $f_{z^*,2n}(z)$, by homomorphically evaluating $f_{z^*,2n}(z)$ with $2n$ homomorphic multiplications and $2n-1$ homomorphic additions.

First, we observe that it is possible to reduce the number of homomorphic multiplications if $N \ll 2n$. Let d be a factor of $2n$ such that $d/2 < N \leq d$. Then, for any $z^*, z \in [N]$, we have that

$$f_{z^*,d}(z) = d^{-1}\sum_{i=0}^{d-1}(x^{\frac{2n}{d}(z^*-z)})^i = \sum_{i=0}^{d-1} c(z,d,i)u^i = \begin{cases} 1, & \text{if } z = z^*; \\ 0, & \text{otherwise,} \end{cases}$$

where $c(z,d,i) = d^{-1}x^{-\frac{2n}{d}zi}, u = x^{\frac{2n}{d}z^*}$. Clearly, given encoding of $u = x^{\frac{2n}{d}z^*}$ for some unknown $z^* \in [N]$ and a public integer $z \in [N]$, the function $f_{z^*,d}(z)$ can be evaluated with at most $d < 2N$ homomorphic multiplications.

Second, we can essentially extend the above technique to $N > 2n$. Let d be a factor of $2n$, and let $e+1 = \lceil \log_d N \rceil \in \mathbb{Z}$. Note that for any integer $z \in [N]$, there exists a unique integer vector $(z_e, \ldots, z_0) \in [d]^{e+1}$ such that $z = \sum_{j=0}^{e} z_j d^j$. Let d_e be a factor of $2n$ such that $d_e/2 \leq \lfloor N/d^e \rfloor < d_e \leq d$. Then, given integers $z^* = \sum_{j=0}^{e} z_j^* d^j, z = \sum_{j=0}^{e} z_j d^j \in [N]$, we have that

$$g_{z^*,N,d}(z) = f_{z_e^*,d_e}(z_e) \cdot \prod_{j=0}^{e-1} f_{z_j^*,d}(z_j) = \begin{cases} 1, & \text{if } z = z^*; \\ 0, & \text{otherwise.} \end{cases}$$

Third, the above technique can be further extended to deal with set inclusion relation. Specifically, given $z^* = \sum_{i=0}^{e} z_j^* d^j \in [N]$ and any set $S \in [N]^\eta$, we have that

$$g_{z^*,N,d}(S) = \sum_{z \in S} g_{z^*,N,d}(z) = \begin{cases} 1, & \text{if } z^* \in S; \\ 0, & \text{otherwise.} \end{cases}$$

Now, we are ready to present our Type-III PHF construction, which is essentially a deterministic algorithm that homomorphically evaluates the function $g_{z^*,N,d}(S)$.

Definition 7. *Let n be a power of 2, and let $q = 3^k$ for some integer k. Let $R_q = \mathbb{Z}_q[x]/(x^n+1)$. For any $\ell = O(n), v = \mathsf{poly}(n) \in \mathbb{Z}$ and $L = 2^\ell$, let $N \leq 16v^2\ell, \eta \leq 4v\ell$ and $CF = \{CF_X\}_{X \in [L]}$ be defined as in Lemma 1. Let $d \geq 2$ be a factor of $2n$, and let $e+1 = \lceil \log_d N \rceil$. Let $d_e \geq 2$ be a factor of $2n$ such that $d_e/2 \leq \lfloor (N-1)/d^e \rfloor < d_e \leq d$. Then, the hash function $\mathcal{H} = (\mathcal{H}.\mathrm{Gen}_R, \mathcal{H}.\mathrm{Eval}_R)$ from $[L]$ to $R_q^{1\times k}$ is defined as follows:*

- $\mathcal{H}.\mathrm{Gen}_R(1^\kappa)$*: Randomly choose $\hat{\mathbf{A}}, \mathbf{A}_i \leftarrow R_q^{1\times k}$ for $0 \leq i \leq e$, return the key $K = (\hat{\mathbf{A}}, \{\mathbf{A}_i\})$.*
- $\mathcal{H}.\mathrm{Eval}_R(K, X)$*: Given the key $K = (\hat{\mathbf{A}}, \{\mathbf{A}_j\})$ and an integer $X \in [L]$ as inputs, compute $\mathbf{U}_z = \mathsf{Rcomp}_{N,d}(\{\mathbf{A}_j\}_{0\leq j\leq e}, d_e, z) \in R_q^{1\times k}$ for all z as shown in* Fig. 2. *Then, return $\mathbf{Z} = \hat{\mathbf{A}} + \sum_{z \in CF_X} \mathbf{U}_z$.*

```
$\mathsf{Rcomp}_{N,d}(\{\mathbf{A}_i\}_{0\le i\le e}, d_e, z)$
Let $z = \sum_{i=0}^{e} z_i d^i$, where $z_i \in [d]$
Let $d_i = d$ for $0 \le i \le e-1$
For $i = 0, \cdots, e$ :
    $\hat{\mathbf{A}}_i = \mathbf{A}_i \cdot x^{-\frac{2n}{d_i} z_i}$
    $\mathbf{U}_{i,d_i-1} = d_i^{-1}\mathbf{G}_3$
    For $j = d_i - 2, \cdots, 0$ :
        $\mathbf{U}_{i,j} = \hat{\mathbf{A}}_i \mathbf{G}_3^{-1}(\mathbf{U}_{i,j+1}) + d_i^{-1}\mathbf{G}_3$
$\mathbf{U}_z = \mathbf{U}_{0,0}$
For $i = 1, \cdots, e$ :
    $\mathbf{U}_z = \mathbf{U}_{i,0}\mathbf{G}_3^{-1}(\mathbf{U}_z)$
Return $\mathbf{U}_z$
```

```
$\mathsf{TRcomp}_{N,d}(\{\mathbf{A}_i, \hat{\mathbf{R}}_i\}_{0\le i\le e}, d_e, z)$
Let $z = \sum_{i=0}^{e} z_i d^i$, where $z_i \in [d]$
Let $d_i = d$ for $0 \le i \le e-1$
For $i = 0, \cdots, e$ :
    $\hat{\mathbf{A}}_i = \mathbf{A}_i \cdot x^{-\frac{2n}{d_i} z_i}$
    $\mathbf{U}_{i,d_i-1} = d_i^{-1}\mathbf{G}_3, \quad \mathbf{R}_{i,d_i-1} = 0, \quad h_{i,d_i-1} = d_i^{-1}$
    For $j = d_i - 2, \cdots, 0$ :
        $\mathbf{U}_{i,j} = \hat{\mathbf{A}}_i \mathbf{G}_3^{-1}(\mathbf{U}_{i,j+1}) + d_i^{-1}\mathbf{G}_3$
        $\mathbf{R}_{i,j} = x^{-\frac{2n}{d_i} z_i}\hat{\mathbf{R}}_i \mathbf{G}_3^{-1}(\mathbf{U}_{i,j+1}) + x^{\frac{2n}{d_i}(z_i^* - z_i)}\mathbf{R}_{i,j+1}$
        $h_{i,j} = x^{\frac{2n}{d_i}(z_i^* - z_i)} h_{i,j+1} + d_i^{-1}$
$\mathbf{U}_z = \mathbf{U}_{0,0}, \mathbf{R}_z = \mathbf{R}_{0,0}, h_z = h_{0,0}$
For $i = 1, \cdots, e$ :
    $\mathbf{U}_z = \mathbf{U}_{i,0}\mathbf{G}_3^{-1}(\mathbf{U}_z)$
    $\mathbf{R}_z = \mathbf{R}_{i,0}\mathbf{G}_3^{-1}(\mathbf{U}_z) + h_{i,0}\mathbf{R}_z, \quad h_z = h_{i,0}h_z$
Return $(\mathbf{R}_z, h_z)$
```

Fig. 2. The Algorithms Used in Definition 7 and Theorem 4

We now show that for any prior fixed $v = \mathsf{poly}(\kappa)$, the hash function $\mathcal{H}$ given in Definition 7 is a $(1, v, \beta)$-PHF for some polynomially bounded $\beta \in \mathbb{R}$.

Theorem 4. *Let n be a power of 2, and let $q = 3^k$ for some integer k. Let $R_q = \mathbb{Z}_q[x]/(x^n+1)$. For any $\ell = O(n), v \le \mathsf{poly}(n) \in \mathbb{Z}$ and $L = 2^\ell$, let $N \le 16v^2\ell, \eta \le 4v\ell$ and $CF = \{CF_X\}_{X\in[L]}$ be defined as in Lemma 1. Let $w = 2\lceil \log_2 q\rceil + 2$. Let $d \ge 2$ be a factor of $2n$, and let $e+1 = \lceil \log_d N\rceil$. Let $d_e \ge 2$ be a factor of $2n$ such that $d_e/2 \le \lfloor (N-1)/d^e \rfloor < d_e \le d$. Then, the hash function $\mathcal{H}$ in Definition 7 is a $(1, v, \beta, \gamma, \delta)$-PHF with $\beta \le v\ell n^{2.5}k^2 ed \cdot \omega(\sqrt{\log n \log nw})$, $\gamma = \mathrm{negl}(\kappa)$ and $\delta = 1/N - \mathrm{negl}(n)$.*

In particular, if we set $\ell = n$ and $d = n^{1/c}$, then $\beta = v \cdot \tilde{O}(n^{3.5+1/c})$, and the key of $\mathcal{H}$ only consists of a constant number $\lceil \log_d N\rceil + 1$ of elements in R_q^k. Moreover, if $\ell = n$ and $v = \omega(\log n)$, then by setting $d = 2n$ we can have that $\beta = \tilde{O}(n^{3.5})$, and that the key of $\mathcal{H}$ only consists of 3 elements in $R_q^{1\times k}$.

Proof. We now construct a pair of trapdoor algorithms for $\mathcal{H}$ as follows:

- $\mathcal{H}.\mathrm{TrapGen}_R(1^\kappa, \mathbf{A}, \mathbf{G}_3)$: Given a uniformly random $\mathbf{A} \leftarrow R_q^{1\times w}$ and matrix $\mathbf{G}_3 \in R_q^{1\times k}$ for $w = 2\lceil \log_2 q\rceil + 2$ as inputs, let $s = \omega(\sqrt{\log nw}) \in \mathbb{R}$ satisfy the requirement in Lemma 9. Randomly choose $\hat{\mathbf{R}}, \hat{\mathbf{R}}_i \leftarrow (D_{R,s})^{w\times k}$ for $0 \le i \le e$, and an integer $z^* = \sum_{i=0}^{e} z_i^* d^i \leftarrow [N]$, where $z_e^* \in [d_e]$ and $z_i^* \in [d]$ for $0 \le i \le e-1$. Then, compute $\hat{\mathbf{A}} = \mathbf{A}\hat{\mathbf{R}} - \mathbf{G}_3$, $\mathbf{A}_i = \mathbf{A}\hat{\mathbf{R}}_i + x^{\frac{2n}{d} z_i^*}\mathbf{G}_3$ for $0 \le i \le e-1$, and $\mathbf{A}_e = \mathbf{A}\hat{\mathbf{R}}_e + x^{\frac{2n}{d_e} z_e^*}\mathbf{G}_3$. Finally, return the key $K' = (\hat{\mathbf{A}}, \{\mathbf{A}_i\})$ and the trapdoor $td = (\hat{\mathbf{R}}, \{\hat{\mathbf{R}}_i\}, z^*)$.
- $\mathcal{H}.\mathrm{TrapEval}_R(td, K', X)$: Given the trapdoor $td = (\hat{\mathbf{R}}, \{\hat{\mathbf{R}}_i\}, z^*)$ for the key $K' = (\hat{\mathbf{A}}, \{\mathbf{A}_i\})$ and an integer $X \in [L]$ as inputs, compute $(\mathbf{R}_z, h_z) =$

$\mathsf{TRcomp}_{N,d}(\{\mathbf{A}_i, \hat{\mathbf{R}}_i\}_{0 \le i \le e}, d_e, z)$ for all $z \in CF_X$ as shown in Fig. 2. Then, return $\mathbf{R}_X = \hat{\mathbf{R}} + \sum_{z \in CF_X} \mathbf{R}_z$ and $\mathbf{S}_X = -1 + \sum_{z \in CF_X} h_z$.

Since $s = \omega(\sqrt{\log nw})$ and $\hat{\mathbf{R}}, \hat{\mathbf{R}}_i \leftarrow (D_{R,s})^{w \times k}$ for $0 \le i \le e$, each vector in the key $K' = (\hat{\mathbf{A}}, \{\mathbf{A}_i\})$ is statistically close to uniform over R_q^k by Lemma 9. Using a standard hybrid argument, it is easy to show that the statistical distance γ between $(\mathbf{A}, K')$ and $(\mathbf{A}, K)$ is negligible, where $K \leftarrow \mathcal{H}.\mathrm{Gen}(1^\kappa)$. In particular, this means that z^* is statistically hidden in K'.

For correctness, it suffices to show that $\mathbf{U}_z = \mathbf{A}\mathbf{R}_z + h_z \mathbf{G}_3$ holds for all $(\mathbf{R}_z, h_z) = \mathsf{TRcomp}_{N,d}(\{\mathbf{A}_i, \hat{\mathbf{R}}_i\}_{0 \le i \le e}, d_e, z)$. First, we show that $\mathbf{U}_{i,0} = \mathbf{A}\mathbf{R}_{i,0} + h_{i,0}\mathbf{G}_3$ always holds for all $0 \le i \le e$. Note that $\mathbf{A}_i = \mathbf{A}\hat{\mathbf{R}}_i + x^{\frac{2n}{d} z_i^*}\mathbf{G}_3$ for $0 \le i \le e-1$ and $\mathbf{A}_e = \mathbf{A}\hat{\mathbf{R}}_e + x^{\frac{2n}{d_e} z_e^*}\mathbf{G}_3$. By the definition of $d_i = d$ for all $0 \le i \le e-1$ and $\hat{\mathbf{A}}_i = \mathbf{A}_i \cdot x^{-\frac{2n}{d_i} z_i}$ for all $0 \le i \le e$, we have that $\hat{\mathbf{A}}_i = \mathbf{A}\hat{\mathbf{R}}_i \cdot x^{-\frac{2n}{d_i} z_i} + x^{\frac{2n}{d_i}(z_i^* - z_i)}\mathbf{G}_3$ for all $0 \le i \le e$. Since we always have $\mathbf{U}_{i,d_i-1} = \mathbf{A}_i\mathbf{R}_{i,d_i-1} + h_{i,d_i-1}\mathbf{G}_3$, by induction it suffices to show that if $\mathbf{U}_{i,j+1} = \mathbf{A}_i\mathbf{R}_{i,j+1} + h_{i,j+1}\mathbf{G}_3$ holds for some $0 \le j \le d_i - 2$, then $\mathbf{U}_{i,j} = \mathbf{A}_i\mathbf{R}_{i,j} + h_{i,j}\mathbf{G}_3$ also holds. Since $\mathbf{U}_{i,j} = \hat{\mathbf{A}}_i\mathbf{G}_3^{-1}(\mathbf{U}_{i,j+1}) + d_i^{-1}\mathbf{G}_3$, we have that $\mathbf{U}_{i,j} = \mathbf{A}(x^{-\frac{2n}{d_i} z_i}\hat{\mathbf{R}}_i\mathbf{G}_3^{-1}(\mathbf{U}_{i,j+1}) + x^{\frac{2n}{d_i}(z_i^* - z_i)}\mathbf{R}_{i,j+1}) + (x^{\frac{2n}{d_i}(z_i^* - z_i)}h_{i,j+1} + d_i^{-1})\mathbf{G}_3 = \mathbf{A}\mathbf{R}_{i,j} + h_{i,j}\mathbf{G}_3$, where $\mathbf{R}_{i,j} = x^{-\frac{2n}{d_i} z_i}\hat{\mathbf{R}}_i\mathbf{G}_3^{-1}(\mathbf{U}_{i,j+1}) + x^{\frac{2n}{d_i}(z_i^* - z_i)}\mathbf{R}_{i,j+1}$ and $h_{i,j} = x^{\frac{2n}{d_i}(z_i^* - z_i)}h_{i,j+1} + d_i^{-1}$. Because $s_1(\mathbf{G}_3^{-1}(\mathbf{U})) \le 3nk$ for any $\mathbf{U} \in R_q^{1 \times k}$, we have that $\mathbf{s}_1(\mathbf{R}_{i,j}) \le 3nk \cdot s_1(\hat{\mathbf{R}}_i) + s_1(\mathbf{R}_{i,j+1})$. By induction, we have that $\mathbf{U}_{i,0} = \mathbf{A}\mathbf{R}_{i,0} + h_{i,0}\mathbf{G}_3$ holds for $s_1(\mathbf{R}_{i,0}) \le 3nk(d_i - 1) \cdot s_1(\hat{\mathbf{R}}_i)$ and $h_{i,0} = d_i^{-1}\sum_{j=0}^{d_i-1} x^{\frac{2n}{d_i}(z_i^* - z_i)j} = f_{z_i^*, d_i}(z_i)$.

Now, we show that $\mathbf{U}_z = \mathbf{A}\mathbf{R}_z + h_z\mathbf{G}_3$ always holds during the second loop. Note that for each $z \in CF_X$, we have that $\mathbf{U}_z = \mathbf{U}_{0,0}, \mathbf{R}_z = \mathbf{R}_{0,0}, h_z = h_{0,0}$ before entering the loop. By the analysis above, we always have that $\mathbf{U}_z = \mathbf{A}\mathbf{R}_z + h_z\mathbf{G}_3$ before entering the second loop, where $h_z = f_{z_0^*, d_0}(z_0)$. By the definition that $\mathbf{U}_z = \mathbf{U}_{i,0}\mathbf{G}_3^{-1}(\mathbf{U}_z)$ and $h_{i,0} \in \{0, 1\}$ we have that $\mathbf{U}_z = (\mathbf{A}\mathbf{R}_{i,0} + h_{i,0}\mathbf{G}_3)\mathbf{G}_3^{-1}(\mathbf{U}_z) = \mathbf{A}(\mathbf{R}_{i,0}\mathbf{G}_3^{-1}(\mathbf{U}_z) + h_{i,0}\mathbf{R}_z) + h_{i,0}h_z\mathbf{G}_3$. This means that $\mathbf{U}_z = \mathbf{A}\mathbf{R}_z + h_z\mathbf{G}_3$ always holds during the computation. By induction, we have that $s_1(\mathbf{R}_z) \le \sum_{i=1}^{e} 3nk \cdot s_1(\mathbf{R}_{i,0}) + s_1(\mathbf{R}_{0,0}) \le 9n^2k^2\sum_{i=1}^{e}(d_i - 1) \cdot s_1(\hat{\mathbf{R}}_i) + 3nk(d_0 - 1) \cdot s_1(\hat{\mathbf{R}}_0)$ and $h_z = \prod_{i=0}^{e} f_{z_i^*, d_i}(z_i) = g_{z^*,N,d}(z)$ after the second loop.

Finally, note that $\hat{\mathbf{A}} = \mathbf{A}\hat{\mathbf{R}} - \mathbf{G}_3$, by the additive homomorphism property, we have that $\mathbf{Z} = \mathbf{A}\mathbf{R}_X + h_X\mathbf{G}_3$ always holds, where $\mathbf{R}_X = \hat{\mathbf{R}} + \sum_{z \in CF_X}\mathbf{R}_z$ and $h_X = -1 + \sum_{z \in CF_X} h_z = -1 + g_{z^*,N,d}(CF_X)$. This means that $s_1(\mathbf{R}_X) \le s_1(\hat{\mathbf{R}}) + |CF_X| \cdot (9n^2k^2\sum_{i=1}^{e}(d_i - 1) \cdot s_1(\hat{\mathbf{R}}_i) + 3nk(d_0 - 1) \cdot s_1(\hat{\mathbf{R}}_0))$. Since $\hat{\mathbf{R}}, \hat{\mathbf{R}}_i \leftarrow D_{R,s}^{w \times k}$ and $w = k = O(\log q) = O(\log n)$, by Lemma 7 we have that $s_1(\hat{\mathbf{R}}), s_1(\hat{\mathbf{R}}_i) \le s\sqrt{n} \cdot O(\sqrt{w} + \sqrt{k} + \omega(\sqrt{\log n})) \le s\sqrt{n} \cdot \omega(\sqrt{\log n})$ holds except with negligible probability. This means that $s_1(\mathbf{R}_X) \le |CF_X| \cdot 9n^{2.5}k^2eds \cdot \omega(\sqrt{\log n})$. Since $|CF_X| \le 4v\ell$ and $s = \omega(\sqrt{\log nw})$, we have that $s_1(\mathbf{R}_X) \le v\ell n^{2.5}k^2ed \cdot \omega(\sqrt{\log n \log nw})$. Besides, because $CF = \{CF_X\}_{X \in [L]}$ is v-cover-free, we have that for any $X_1, Y_1, \ldots, Y_v \in [L]$ such that $X_1 \ne Y_j$ for all $j \in \{1, \ldots, v\}$, there is at least one element in $CF_{X_1} \subseteq [N]$ that does not belong to the union set $\cup_{j\in\{1,\ldots,v\}} CF_{Y_j}$. Since z^* is randomly chosen from $[N]$ and is statistically hidden in the key K', the probability $\Pr[z^* \in CF_{X_1} \wedge z^* \notin \cup_{j\in\{1,\ldots,v\}} CF_{Y_j}]$ is at least $1/N - \mathrm{negl}(n)$. Since $h_X = -1 + g_{z^*,N,d}(CF_X)$, which is equal to 0 if $z^* \in CF_X$ and -1 otherwise, we have that $\Pr[h_{X_1} = 0 \wedge h_{Y_1} = \cdots = h_{Y_v} = -1] \geq 1/N - \mathrm{negl}(n)$.

The second claim follows from the fact that $\log_d N$ is a constant for prior fixed $v = \mathsf{poly}(n)$. The third claim follows from the fact that $s_1(\mathbf{R}_X) \leq s_1(\hat{\mathbf{R}}) + |CF_X| \cdot (9n^2k^2 \sum_{i=1}^{e}(d_i - 1) \cdot s_1(\hat{\mathbf{R}}_i) + 3nk(d_0 - 1) \cdot s_1(\hat{\mathbf{R}}_0)) = \tilde{O}(n^{3.5})$ for the setting of $e = 1, d_0 = 2n$ and $d_1 = \omega(\log n)$. □

3.4 Improved Type-III Construction for $v < n$

In this section, we present an improved Type-III PHF construction with constant keys over rings, which allows us to obtain better parameters by using another way to compute the function $g_{z^*,N,d}(CF_X)$. This improvement is basically due to the asymmetry between homomorphic multiplications and homomorphic additions.

Recall that given integers $z^* = \sum_{j=0}^{e} z_j^* d^j, z = \sum_{j=0}^{e} z_j d^j \in [N]$, we have that

$$g_{z^*,N,d}(z) = f_{z_e^*,d_e}(z_e) \cdot \prod_{j=0}^{e-1} f_{z_j^*,d}(z_j) = \begin{cases} 1, & \text{if } z = z^*; \\ 0, & \text{otherwise.} \end{cases}$$

By using the fact that $f_{z_j^*,d}(z_j) = \sum_{i=0}^{d-1} c(z_j, d, i) u^i$, where $c(z_j, d, i) = d^{-1} x^{-\frac{2n}{d} z_j i}$, $u = x^{\frac{2n}{d} z_j^*}$. We can rewrite the function $g_{z^*,N,d}(z)$ as

$$\begin{aligned} g_{z^*,N,d}(z) &= f_{z_e^*,d_e}(z_e) \cdot \prod_{j=0}^{e-1} f_{z_j^*,d}(z_j) \\ &= d_e^{-1} \sum_{i_e=0}^{d_e-1} (x^{\frac{2n}{d_e}(z_e^* - z_e)})^{i_e} \cdot \prod_{j=0}^{e-1} \left(d^{-1} \sum_{i_j=0}^{d-1} (x^{\frac{2n}{d}(z_j^* - z_j)})^{i_j} \right) \\ &= \sum_{i_e=0}^{d_e-1} c(z_e, d_e, i_e) x^{\frac{2n}{d_e} z_e^* i_e} \cdot \prod_{j=0}^{e-1} \left(\sum_{i_j=0}^{d-1} c(z_j, d, i_j) x^{\frac{2n}{d} z_j^* i_j} \right) \\ &= \sum_{i_e=0}^{d_e-1} \left(\sum_{i_{e-1}=0}^{d-1} \left(\cdots \left(\sum_{i_0=0}^{d-1} C(z, N, d, i) x^{\frac{2n}{d} z_0^* i_0} \right) \cdots \right) x^{\frac{2n}{d} z_{e-1}^* i_{e-1}} \right) x^{\frac{2n}{d_e} z_e^* i_e} \end{aligned}$$

where $C(z, N, d, i) = c(z_e, d_e, i_e) \prod_{j=0}^{e-1} c(z_j, d, i_j) = d_e^{-1} d^{-e} x^{-\frac{2n}{d_e} z_e i_e - \frac{2n}{d} \sum_{j=0}^{e-1} z_j i_j}$ and $i = \sum_{j=0}^{e} i_j d^j$. Thus, given encodings of $x^{\frac{2n}{d_e} z_e^*}$ and $\{x^{\frac{2n}{d} z_j^*}\}_{j=0}^{e-1}$, one can evaluate $g_{z^*,N,d}(z)$ by using $d_e d^e < 2N$ homomorphic multiplications.

Similarly, given $z^* = \sum_{i=0}^{e} z_j^* d^j \in [N]$ and any set $S \in [N]^\eta$, we an rewrite the function

$$g_{z^*,N,d}(S) = \sum_{z \in S} g_{z^*,N,d}(z) = \begin{cases} 1, & \text{if } z^* \in S; \\ 0, & \text{otherwise.} \end{cases}$$

as

$$\begin{aligned}
g_{z^*,N,d}(S) &= \sum_{z\in S} g_{z^*,N,d}(z)\\
&= \sum_{z\in S}\sum_{i_e=0}^{d_e-1}\left(\sum_{i_{e-1}=0}^{d-1}\left(\cdots\left(\sum_{i_0=0}^{d-1} C(z,N,d,i)x^{\frac{2n}{d}z_0^* i_0}\right)\cdots\right)x^{\frac{2n}{d}z_{e-1}^* i_{e-1}}\right)x^{\frac{2n}{d_e}z_e^* i_e}\\
&= \sum_{i_e=0}^{d_e-1}\left(\sum_{i_{e-1}=0}^{d-1}\left(\cdots\left(\sum_{i_0=0}^{d-1}\sum_{z\in S} C(z,N,d,i)x^{\frac{2n}{d}z_0^* i_0}\right)\cdots\right)x^{\frac{2n}{d}z_{e-1}^* i_{e-1}}\right)x^{\frac{2n}{d_e}z_e^* i_e}\\
&= \sum_{i_e=0}^{d_e-1}\left(\sum_{i_{e-1}=0}^{d-1}\left(\cdots\left(\sum_{i_0=0}^{d-1} C(S,N,d,i)x^{\frac{2n}{d}z_0^* i_0}\right)\cdots\right)x^{\frac{2n}{d}z_{e-1}^* i_{e-1}}\right)x^{\frac{2n}{d_e}z_e^* i_e}
\end{aligned}$$

where $C(S,N,d,i) = \sum_{z\in S} C(z,N,d,i)$ and $i = \sum_{j=0}^{e} i_j d^j$. Clearly, given encodings of $x^{\frac{2n}{d_e}z_e^*}$ and $\{x^{\frac{2n}{d}z_j^*}\}_{j=0}^{e-1}$, we can still evaluate $g_{z^*,N,d}(S)$ by using $d_e d^e < 2N$ homomorphic multiplications. Now, we give our improved Type-III PHF construction.

Definition 8. *Let n be a power of 2, and let $q = 3^k$ for some integer k. Let $R_q = \mathbb{Z}_q[x]/(x^n+1)$. For any $\ell = O(n), v = \mathsf{poly}(n) \in \mathbb{Z}$ and $L = 2^\ell$, let $N \leq 16v^2\ell, \eta \leq 4v\ell$ and $CF = \{CF_X\}_{X\in[L]}$ be defined as in Lemma 1. Let $d = 2n$, and let $e+1 = \lceil \log_d N\rceil$. Let $d_e \geq 2$ be a factor of $2n$ such that $d_e/2 \leq \lfloor (N-1)/d^e \rfloor < d_e$. Then, the hash function $\mathcal{H} = (\mathcal{H}.\mathrm{Gen}_R, \mathcal{H}.\mathrm{Eval}_R)$ from $[L]$ to $R_q^{1\times k}$ is defined as follows:*

- $\mathcal{H}.\mathrm{Gen}_R(1^\kappa)$*: Randomly choose $\hat{\mathbf{A}}, \mathbf{A}_i \leftarrow R_q^{1\times k}$ for $0 \leq i \leq e$, return the key $K = (\hat{\mathbf{A}}, \{\mathbf{A}_i\})$.*
- $\mathcal{H}.\mathrm{Eval}_R(K,X)$*: Given the key $K = (\hat{\mathbf{A}}, \{\mathbf{A}_i\})$ and an integer $X \in [L]$ as inputs, compute $\hat{\mathbf{U}}_X = \mathsf{rec\text{-}Rcomp}_{K,X}(e,0,d_e) \in R_q^{1\times k}$ as shown in* Fig. 3*, and return $\mathbf{Z} = \hat{\mathbf{A}} + \hat{\mathbf{U}}_X$.*

We now show that for any prior fixed $v = \mathsf{poly}(\kappa)$, the hash function $\mathcal{H}$ given in Definition 8 is a $(1, v, \beta)$-PHF for some polynomially bounded $\beta \in \mathbb{R}$.

Theorem 5. *Let n be a power of 2, and let $q = 3^k$ for some integer k. Let $R_q = \mathbb{Z}_q[x]/(x^n+1)$. For any $\ell = O(n), v \leq \mathsf{poly}(n) \in \mathbb{Z}$ and $L = 2^\ell$, let $N \leq 16v^2\ell, \eta \leq 4v\ell$ and $CF = \{CF_X\}_{X\in[L]}$ be defined as in Lemma 1. Let $w = 2\lceil \log_2 q\rceil + 2$. Let $d = 2n$, and let $e+1 = \lceil \log_d N\rceil$. Let $d_e \geq 2$ be a factor of $2n$ such that $d_e/2 \leq \lfloor (N-1)/d^e \rfloor < d_e$. Then, the hash function $\mathcal{H}$ in Definition 8 is a $(1, v, \beta, \gamma, \delta)$-PHF with $\beta \leq v^2 n^{1.5}\ell k \cdot \omega(\sqrt{\log n \log nw})$, $\gamma = \mathrm{negl}(\kappa)$ and $\delta = 1/N - \mathrm{negl}(n)$.*

In particular, if we set $\ell = n$ and $v = \omega(\log n)$, then $\beta = \tilde{O}(n^{2.5})$, and the key of $\mathcal{H}$ only consists of 3 elements in $R_q^{1\times k}$.

```
$\mathsf{rec\text{-}Rcomp}_{K,X}(r,i,t)$ :
  If $r = 0$ :
    $\mathbf{U}_{r,i,t-1} = C(CF_X, N, d, di+t-1)\mathbf{G}_3$
    For $j = t-2, \cdots, 0$ :
      $\mathbf{U}_{r,i,j} = \mathbf{A}_r\mathbf{G}_3^{-1}(\mathbf{U}_{r,i,j+1}) + C(CF_X, N, d, di+j)\mathbf{G}_3$
  Else:
    $\mathbf{U}_{r,i,t-1} = \mathsf{rec\text{-}Rcomp}_{K,X}(r-1, di+t-1, d)$
    For $j = t-2, \cdots, 0$ :
      $\mathbf{U}_{r,di+j,0} = \mathsf{rec\text{-}Rcomp}_{K,X}(r-1, di+j, d)$
      $\mathbf{U}_{r,i,j} = \mathbf{A}_r\mathbf{G}_3^{-1}(\mathbf{U}_{r,i,j+1}) + \mathbf{U}_{r,di+j,0}$
  Return $\mathbf{U}_{r,i,0}$
```

```
$\mathsf{rec\text{-}TRcomp}_{td,K,X}(r,i,t)$ :
  If $r = 0$ :
    $\mathbf{U}_{r,i,t-1} = C(CF_X, N, d, di+t-1)\mathbf{G}_3$
    $\mathbf{R}_{r,i,t-1} = 0, \quad h_{r,i,t-1} = C(CF_X, N, d, di+t-1)$
    For $j = t-2, \cdots, 0$ :
      $\mathbf{U}_{r,i,j} = \mathbf{A}_r\mathbf{G}_3^{-1}(\mathbf{U}_{r,i,j+1}) + C(CF_X, N, d, di+j)\mathbf{G}_3$
      $\mathbf{R}_{r,i,j} = \hat{\mathbf{R}}_r\mathbf{G}_3^{-1}(\mathbf{U}_{r,i,j+1}) + \mathbf{R}_{r,i,j+1}x^{\frac{2n}{t}z_r^*}$
      $h_{r,i,j} = h_{r,i,j+1}x^{\frac{2n}{t}z_r^*} + C(CF_X, N, d, di+j)$
  Else:
    $(\mathbf{U}_{r,i,t-1}, \mathbf{R}_{r,i,t-1}, h_{r,i,t-1}) = \mathsf{rec\text{-}TRcomp}_{td,K,X}(r-1, di+t-1, d)$
    For $j = t-2, \cdots, 0$ :
      $(\mathbf{U}_{r-1,di+j,0}, \mathbf{R}_{r-1,di+j,0}, h_{r-1,di+j,0}) = \mathsf{rec\text{-}TRcomp}_{td,K,X}(r-1, di+j, d)$
      $\mathbf{U}_{r,i,j} = \mathbf{A}_r\mathbf{G}_3^{-1}(\mathbf{U}_{r,i,j+1}) + \mathbf{U}_{r-1,di+j,0}$
      $\mathbf{R}_{r,i,j} = \hat{\mathbf{R}}_r\mathbf{G}_3^{-1}(\mathbf{U}_{r,i,j+1}) + \mathbf{R}_{r,i,j+1}x^{\frac{2n}{t}z_r^*} + \mathbf{R}_{r-1,di+j,0}$
      $h_{r,i,j} = h_{r,i,j+1}x^{\frac{2n}{t}z_r^*} + h_{r-1,di+j,0}$
  Return $(\mathbf{U}_{r,i,0}, \mathbf{R}_{r,i,0}, h_{r,i,0})$
```

Fig. 3. The Recursive Algorithms Used in Definition 8 and Theorem 5

Proof. We now construct a pair of trapdoor algorithms for $\mathcal{H}$ as follows:

- $\mathcal{H}.\mathrm{TrapGen}_R(1^\kappa, \mathbf{A}, \mathbf{G}_3)$: Given a uniformly random $\mathbf{A} \leftarrow R_q^{1\times w}$ and matrix $\mathbf{G}_3 \in R_q^{1\times k}$ for $w = 2\lceil \log_2 q\rceil + 2$ as inputs, let $s = \omega(\sqrt{\log nw}) \in \mathbb{R}$ satisfy the requirement in Lemma 9. Randomly choose $\hat{\mathbf{R}}, \hat{\mathbf{R}}_i \leftarrow (D_{R,s})^{w\times k}$ for $0 \le i \le e$, and an integer $z^* = \sum_{i=0}^{e} z_i^* d^i \leftarrow [N]$, where $z_e^* \in [d_e]$ and $z_i^* \in [d]$ for $0 \le i \le e-1$. Then, compute $\hat{\mathbf{A}} = \mathbf{A}\hat{\mathbf{R}} - \mathbf{G}_3, \mathbf{A}_i = \mathbf{A}\hat{\mathbf{R}}_i + x^{\frac{2n}{d}z_i^*}\mathbf{G}_3$ for $0 \le i \le e-1$, and $\mathbf{A}_e = \mathbf{A}\hat{\mathbf{R}}_e + x^{\frac{2n}{d_e}z_e^*}\mathbf{G}_3$. Finally, return the key $K' = (\hat{\mathbf{A}}, \{\mathbf{A}_i\})$ and the trapdoor $td = (\hat{\mathbf{R}}, \{\mathbf{R}_i\}, z^*)$.
- $\mathcal{H}.\mathrm{TrapEval}_R(td, K', X)$: Given the trapdoor $td = (\hat{\mathbf{R}}, \{\mathbf{R}_i\}, z^*)$ for the key $K' = (\hat{\mathbf{A}}, \{\mathbf{A}_i\})$ and an integer $X \in [L]$ as inputs, compute $(\hat{\mathbf{U}}_X, \hat{\mathbf{R}}_X, \hat{h}_X) = \mathsf{rec\text{-}TRcomp}_{td,K',X}(e, 0, d_e) \in R_q^k$ as shown in Fig. 3, and return $(\mathbf{R}_X, h_X) = (\hat{\mathbf{R}} + \hat{\mathbf{R}}_X, -1 + \hat{h}_X) \in R_q^{w\times k} \times R_q$.

Since $s = \omega(\sqrt{\log nw})$ and $\hat{\mathbf{R}}, \hat{\mathbf{R}}_i \leftarrow (D_{R,s})^{w\times k}$ for $0 \le i \le e$, each vector in the key $K' = (\hat{\mathbf{A}}, \{\mathbf{A}_i\})$ is statistically close to uniform over $R_q^{1\times k}$ by Lemma 9. Using a standard hybrid argument, it is easy to show that the statistical distance γ between $(\mathbf{A}, K')$ and $(\mathbf{A}, K)$ is negligible, where $K \leftarrow \mathcal{H}.\mathrm{Gen}(1^\kappa)$. In particular, this means that z^* is statistically hidden in K'.

For correctness, it is easy to check that given the same inputs (td, K', X), both $\mathsf{rec\text{-}Rcomp}_{K',X}(e, 0, d_e)$ and $\mathsf{rec\text{-}TRcomp}_{td,K',X}(e, 0, d_e)$ essentially compute $\mathbf{U}_{r,i,j}$ the same way for all possible (r, i, j), and thus will output the same $\mathbf{U}_{e,0,0}$. This means that we only have to show that $\hat{\mathbf{U}}_X = \mathbf{A}\hat{\mathbf{R}}_X + \hat{h}_X\mathbf{G}_3 = \mathbf{A}\mathbf{R}_{e,0,0} + h_{e,0,0}\mathbf{G}_3 = \mathbf{U}_{e,0,0}$ holds for all $(\mathbf{U}_{e,0,0}, \mathbf{R}_{e,0,0}, h_{e,0,0}) = \mathsf{rec\text{-}TRcomp}_{td,K',X}(e, 0, d_e)$. We prove this claim by induction.

First, we show that $\mathbf{U}_{0,i,0} = \mathbf{A}\mathbf{R}_{0,i,0} + h_{0,i,0}\mathbf{G}_3$ always holds for all possible choices of (i, t) and $(\mathbf{U}_{0,i,0}, \mathbf{R}_{0,i,0}, h_{0,i,0}) = \mathsf{rec\text{-}TRcomp}_{td,K',X}(0, i, t)$. Note that we always have $\mathbf{U}_{0,i,t-1} = \mathbf{A}\mathbf{R}_{0,i,t-1} + h_{0,i,t-1}\mathbf{G}_3$ during the computation of $\mathsf{rec\text{-}TRcomp}_{td,K',X}(0, i, t)$. By induction it suffices to show that if $\mathbf{U}_{0,i,j+1} = \mathbf{A}\mathbf{R}_{0,i,j+1} + h_{0,i,j+1}\mathbf{G}_3$ holds for some $0 \le j \le t-2$, then $\mathbf{U}_{0,i,j} = \mathbf{A}\mathbf{R}_{0,i,j} +$

$h_{0,i,j}\mathbf{G}_3$ also holds. Since $\mathbf{U}_{0,i,j} = \mathbf{A}_0 \cdot \mathbf{G}_3^{-1}(\mathbf{U}_{0,i,j+1}) + C(CF_X, N, d, di+j)\mathbf{G}_3$ and $\mathbf{A}_0 = \mathbf{A}\hat{\mathbf{R}}_0 + x^{\frac{2n}{t}z_0^*}\mathbf{G}_3$, we have that $\mathbf{U}_{0,i,j} = \mathbf{A}(\hat{\mathbf{R}}_0\mathbf{G}_3^{-1}(\mathbf{U}_{0,i,j+1}) + \mathbf{R}_{0,i,j+1}x^{\frac{2n}{t}z_0^*}) + (h_{0,i,j+1}x^{\frac{2n}{t}z_0^*} + C(CF_X, N, d, di+j))\mathbf{G}_3$, which is equivalent to $\mathbf{A}\mathbf{R}_{0,i,j} + h_{0,i,j}\mathbf{G}_3$, where $\mathbf{R}_{0,i,j} = \hat{\mathbf{R}}_0\mathbf{G}_3^{-1}(\mathbf{U}_{0,i,j+1}) + \mathbf{R}_{0,i,j+1}x^{\frac{2n}{t}z_0^*}$ and $h_{0,i,j} = h_{0,i,j+1}x^{\frac{2n}{t}z_0^*} + C(CF_X, N, d, di+j)$. Because $s_1(\mathbf{G}_3^{-1}(\mathbf{U})) \leq 3nk$ for any $\mathbf{U} \in R_q^{1\times k}$, we have that $s_1(\mathbf{R}_{0,i,j}) \leq 3nk \cdot s_1(\hat{\mathbf{R}}_0) + s_1(\mathbf{R}_{0,i,j+1})$. By induction, we have that $\mathbf{U}_{0,i,0} = \mathbf{A}\mathbf{R}_{0,i,0} + h_{0,i,0}\mathbf{G}_3$ holds for $s_1(\mathbf{R}_{0,i,0}) \leq 3nk(t-1)\cdot s_1(\hat{\mathbf{R}}_0)$ and $h_{0,i,0} = \sum_{j=0}^{t-1} C(CF_X, N, d, di+j)x^{\frac{2n}{t}z_0^* j}$.

Now, we show that if there exists some $1 \leq r \leq e$ such that $\mathbf{U}_{r-1,i',0} = \mathbf{A}\mathbf{R}_{r-1,i',0} + h_{r-1,i',0}\mathbf{G}_3$ holds for all possible choices of (i', t') and $(\mathbf{U}_{r-1,i',0}, \mathbf{R}_{r-1,i',0}, h_{r-1,i',0}) = \mathsf{rec\text{-}TRcomp}_{td,K',X}(r-1, i', t')$, then we have that $\mathbf{U}_{r,i,0} = \mathbf{A}\mathbf{R}_{r,i,0} + h_{r,i,0}\mathbf{G}_3$ holds for all possible choices of (i, t) and $(\mathbf{U}_{r,i,0}, \mathbf{R}_{r,i,0}, h_{r,i,0}) = \mathsf{rec\text{-}TRcomp}_{td,K',X}(r, i, t)$. By definition we have $(\mathbf{U}_{r,i,t-1}, \mathbf{R}_{r,i,t-1}, h_{r,i,t-1}) = (\mathbf{U}_{r-1,di+t-1,0}, \mathbf{R}_{r-1,di+t-1,0}, h_{r-1,di+t-1,0}) = \mathsf{rec\text{-}TRcomp}_{td,K',X}(r-1, di+t-1, d)$. This means that $\mathbf{U}_{r,i,t-1} = \mathbf{A}\mathbf{R}_{r,i,t-1} + h_{r,i,t-1}\mathbf{G}_3$ and that $\mathbf{s}_1(\mathbf{R}_{r,i,t-1}) = \mathbf{s}_1(\mathbf{R}_{r-1,di+t-1,0})$. Similarly, we have $\mathbf{U}_{r-1,di+j,0} = \mathbf{A}\hat{\mathbf{R}}_{r-1,di+j,0} + h_{r-1,di+j,0}\mathbf{G}_3$ for all $0 \leq j \leq t-2$ by assumption. Since $\mathbf{U}_{r,i,j} = \mathbf{A}_r\mathbf{G}_3^{-1}(\mathbf{U}_{r,i,j+1}) + \mathbf{U}_{r-1,di+j,0}$, we have that $\mathbf{U}_{r,i,j} = \mathbf{A}(\hat{\mathbf{R}}_r\mathbf{G}_3^{-1}(\mathbf{U}_{r,i,j+1}) + \mathbf{R}_{r,i,j+1}x^{\frac{2n}{t}z_r^*} + \mathbf{R}_{r-1,di+j,0}) + (h_{r,i,j+1}x^{\frac{2n}{t}z_r^*} + h_{r-1,di+j,0})\mathbf{G}_3$, which is equivalent to $\mathbf{A}\mathbf{R}_{r,i,j} + h_{r,i,j}\mathbf{G}_3$, where $\mathbf{R}_{r,i,j} = \hat{\mathbf{R}}_r\mathbf{G}_3^{-1}(\mathbf{U}_{r,i,j+1}) + \mathbf{R}_{r,i,j+1}x^{\frac{2n}{t}z_r^*} + \mathbf{R}_{r-1,di+j,0}$ and $h_{r,i,j} = h_{r,i,j+1}x^{\frac{2n}{t}z_r^*} + h_{r-1,di+j,0}$. Note that $s_1(\mathbf{R}_{r,i,j}) \leq 3nk \cdot s_1(\hat{\mathbf{R}}_r) + s_1(\mathbf{R}_{r,i,j+1}) + s_1(\mathbf{R}_{r-1,di+j,0})$. By induction, we have that $\mathbf{U}_{r,i,0} = \mathbf{A}\mathbf{R}_{r,i,0} + h_{r,i,0}\mathbf{G}_3$ holds for $s_1(\mathbf{R}_{r,i,0}) \leq 3nk(t-1)s_1(\hat{\mathbf{R}}_r) + \sum_{j=0}^{t-1} s_1(\mathbf{R}_{r-1,di+j,0})$, and $h_{r,i,0} = \sum_{j=0}^{t-1} h_{r-1,di+j,0}(x^{\frac{2n}{t}z_r^*})^j$.

In all, for any $(\mathbf{U}_{e,0,0}, \mathbf{R}_{e,0,0}, h_{e,0,0}) = \mathsf{rec\text{-}TRcomp}_{td,K',X}(e, 0, d_e)$, we have that $\mathbf{U}_{e,0,0} = \mathbf{A}\mathbf{R}_{e,0,0} + h_{e,0,0}\mathbf{G}_3$ holds, where $s_1(\mathbf{R}_{e,0,0}) = 3nk(d_e - 1)s_1(\hat{\mathbf{R}}_e) + \sum_{j=1}^{e} 3nkd_e d^{j-1}(d-1)s_1(\hat{\mathbf{R}}_{e-j})$ and $h_{e,0,0} = g_{z^*,N,d}(CF_X)$. Since $\hat{\mathbf{R}}, \hat{\mathbf{R}}_i \leftarrow D_{R,s}^{w\times k}$ and $w = k = O(\log q) = O(\log n)$, by Lemma 7 we have $s_1(\hat{\mathbf{R}}), s_1(\hat{\mathbf{R}}_i) \leq s\sqrt{n} \cdot O(\sqrt{w} + \sqrt{k} + \omega(\sqrt{\log n})) \leq s\sqrt{n} \cdot \omega(\sqrt{\log n})$ except with negligible probability. This means that $s_1(\mathbf{R}_{e,0,0}) \leq 3n^{1.5}k(2N-1)s \cdot \omega(\sqrt{\log n}) < 6n^{1.5}kNs \cdot \omega(\sqrt{\log n})$. Since $N \leq 16v^2\ell$ and $s = \omega(\sqrt{\log nw})$, we have $s_1(\mathbf{R}_{e,0,0}) \leq v^2n^{1.5}\ell k \cdot \omega(\sqrt{\log n \log nw})$. Similarly, because $CF = \{CF_X\}_{X\in[L]}$ is v-cover-free, for any $X_1, Y_1, \ldots, Y_v \in [L]$ such that $X_1 \neq Y_j$ for all $j \in \{1, \ldots, v\}$, there is at least one element in $CF_{X_1} \subseteq [N]$ that does not belong to the union set $\cup_{j\in\{1,\ldots,v\}}CF_{Y_j}$. Since z^* is randomly chosen from $[N]$ and is statistically hidden in the key K', the probability $\Pr[z^* \in CF_{X_1} \wedge z^* \notin \cup_{j\in\{1,\ldots,v\}}CF_{Y_j}]$ is at least $1/N - \mathrm{negl}(n)$. Since $h_{e,0,0} = g_{z^*,N,d}(CF_X)$, which is equal to 1 if $z^* \in CF_X$ and 0 otherwise, we have $\Pr[h_{X_1} = 0 \wedge h_{Y_1} = \cdots = h_{Y_v} = 1] \geq 1/N - \mathrm{negl}(n)$.

The second claim follows from the fact that $\ell = n$ and $v = \omega(\log n)$. □

3.5 Collision-Resistance and High Min-Entropy

Collision-Resistance. Let $\mathcal{H} = \{\mathrm{H}_K : \mathcal{X} \to \mathcal{Y}\}_{K\in\mathcal{K}}$ be a family of hash functions with key space $\mathcal{K}$. We say that $\mathcal{H}$ is collision-resistant if for any PPT algorithm $\mathcal{C}$, its advantage

$$\mathrm{Adv}^{\mathrm{cr}}_{\mathcal{H},\mathcal{C}}(\kappa) = \Pr[K \leftarrow \mathcal{K}; (X_1, X_2) \leftarrow \mathcal{C}(K, 1^\kappa) : X_1 \neq X_2 \wedge \mathrm{H}_K(X_1) = \mathrm{H}_K(X_2)]$$

is negligible in the security parameter κ.

Theorem 6. *Let $n, \bar{n}, v, q \in \mathbb{Z}$ and $\bar{\beta}, \beta \in \mathbb{R}$ be polynomials in the security parameter κ. Let $\mathcal{H} = (\mathcal{H}.\mathrm{Gen}_\mathcal{R}, \mathcal{H}.\mathrm{Eval}_\mathcal{R})$ be a $(1, v, \beta, \gamma, \delta)$-PHF defined over $\mathcal{R}$ with $\gamma = \mathrm{negl}(\kappa)$ and noticeable $\delta > 0$. Then, for large enough $\bar{m}, m \in \mathbb{Z}$ and $v \geq 1$, if there exists an algorithm $\mathcal{C}$ breaking the collision-resistance of $\mathcal{H}$, there exists an algorithm $\mathcal{B}$ solving the $\mathrm{ISIS}_{q,\bar{m},\bar{\beta}}$ problem over $\mathcal{R}$ for $\bar{\beta} = \beta\sqrt{mn} \cdot \omega(\sqrt{\log n\bar{n}})$ with probability at least $\epsilon' \geq (\epsilon - \gamma)\delta$, where $n = 1$ if $\mathcal{R} = \mathbb{Z}$.*

Proof. If there exists an algorithm $\mathcal{C}$ breaking the collision-resistance of $\mathcal{H}$ with advantage ϵ, we now construct an algorithm $\mathcal{B}$ that solves the $\mathrm{ISIS}_{q,\bar{m},\bar{\beta}}$ problem. Let $\mathbf{B} \in \mathcal{R}_q^{\bar{n}\times m}$ be any trapdoor matrix that allows to efficiently sample short vector $\mathbf{v} \in \mathcal{R}^m$ such that $\|\mathbf{v}\| \leq \sqrt{mn} \cdot \omega(\sqrt{\log n\bar{n}})$ and $\mathbf{B}\mathbf{v} = \mathbf{u}'$ for any $\mathbf{u}' \in \mathcal{R}_q^{\bar{n}}$. Formally, given an $\mathrm{ISIS}_{q,\bar{m},\bar{\beta}}$ challenge instance $(\mathbf{A}, \mathbf{u}) \in \mathcal{R}_q^{\bar{n}\times\bar{m}} \times \mathcal{R}_q^{\bar{n}}$. The algorithm $\mathcal{B}$ computes $(K', td) \leftarrow \mathcal{H}.\mathrm{TrapGen}_\mathcal{R}(1^\kappa, \mathbf{A}, \mathbf{B})$, and sends K' as the hash key to $\mathcal{C}$. Since the statistical distance between K' and the real hash key K is at most $\gamma = \mathrm{negl}(\kappa)$, the probability that given the key K' the algorithm $\mathcal{C}(K', 1^\kappa)$ outputs two elements $X_1 \neq X_2$ satisfying $\mathrm{H}_{K'}(X_1) = \mathrm{H}_{K'}(X_2)$, is at least $\epsilon - \gamma$. By the correctness of $\mathcal{H}$, we know that there exist two tuples $(\mathbf{R}_{X_1}, \mathbf{S}_{X_1})$ and $(\mathbf{R}_{X_2}, \mathbf{S}_{X_2})$ such that $\mathrm{H}_{K'}(X_1) = \mathbf{A}\mathbf{R}_{X_1} + \mathbf{S}_{X_1}\mathbf{B} = \mathbf{A}\mathbf{R}_{X_2} + \mathbf{S}_{X_2}\mathbf{B} = \mathrm{H}_{K'}(X_2)$. In addition, by the well-distributed hidden matrices property of $\mathcal{H}$, the probability $\Pr[\mathbf{S}_{X_1} = \mathbf{0} \wedge \mathbf{S}_{X_2} \in \mathcal{I}_n]$ is at least δ. In other words, the equation $\mathbf{A}\mathbf{R}_{X_1} = \mathbf{A}\mathbf{R}_{X_2} + \mathbf{S}_{X_2}\mathbf{B}$ holds with probability at least $(\epsilon - \gamma)\delta$. If this is the case, $\mathcal{B}$ outputs $\mathbf{x} = (\mathbf{R}_{X_1} - \mathbf{R}_{X_2})\mathbf{v}$, where $\mathbf{v} \in \mathcal{R}_q^m$ is sampled by using the trapdoor of $\mathbf{B}$ such that $\|\mathbf{v}\| \leq \sqrt{mn} \cdot \omega(\sqrt{\log n\bar{n}})$ and $\mathbf{B}\mathbf{v} = \mathbf{S}_{X_2}^{-1}\mathbf{u}$. By $\mathbf{A}\mathbf{x} = \mathbf{S}_{X_2}\mathbf{B}\mathbf{v} = \mathbf{u}$, we have that $\mathbf{x}$ is a solution of $\mathbf{A}\mathbf{x} = \mathbf{u}$. In addition, since $s_1(\mathbf{R}_{X_1}), s_1(\mathbf{R}_{X_2}) \leq \beta$ by assumption, we have $\|\mathbf{x}\| \leq \beta\sqrt{mn} \cdot \omega(\sqrt{\log n\bar{n}})$. This completes the proof. □

High Min-Entropy. Let $\mathcal{H} : \mathcal{X} \to \mathcal{R}_q^{\bar{n}\times m}$ be a $(1, v, \beta, \gamma, \delta)$-PHF with $\gamma = \mathrm{negl}(\kappa)$ and noticeable $\delta > 0$. Note that the well-distributed hidden matrices property of $\mathcal{H}$ holds even for an unbounded algorithm $\mathcal{A}$ that chooses $\{X_i\}$ and $\{Y_j\}$ after seeing K'. For any noticeable $\delta > 0$, this can only happen when the decomposition $\mathrm{H}_{K'}(X) = \mathbf{A}\mathbf{R}_X + \mathbf{S}_X\mathbf{B}$ is not unique (with respect to K') and the particular pair determined by td, i.e., $(\mathbf{R}_X, \mathbf{S}_X) = \mathcal{H}.\mathrm{TrapEval}(td, K', X)$, is information-theoretically hidden from $\mathcal{A}$. We now introduce a property called high min-entropy to formally capture this useful feature.

Definition 9 (PHF with High Min-Entropy). *Let $\mathcal{H} : \mathcal{X} \to \mathcal{R}_q^{\bar{n}\times m}$ be a $(1, v, \beta, \gamma, \delta)$-PHF with $\gamma = \mathrm{negl}(\kappa)$ and noticeable $\delta > 0$. Let $\mathcal{K}$ be the key space of $\mathcal{H}$, and let $\mathcal{H}.\mathrm{TrapGen}_\mathcal{R}$ and $\mathcal{H}.\mathrm{TrapEval}_\mathcal{R}$ be a pair of trapdoor generation* *and trapdoor evaluation algorithms for $\mathcal{H}$. We say that $\mathcal{H}$ is a* PHF *with high min-entropy if for uniformly random $\mathbf{A} \in \mathcal{R}_q^{\bar{n}\times\bar{m}}$ and (publicly known) trapdoor matrix $\mathbf{B} \in \mathcal{R}_q^{\bar{n}\times m}$, the following conditions hold*

1. *For any $(K', td) \leftarrow \mathcal{H}.\mathrm{TrapGen}_{\mathcal{R}}(1^\kappa, \mathbf{A}, \mathbf{B}), K \leftarrow \mathcal{H}.\mathrm{Gen}_{\mathcal{R}}(1^\kappa)$, any $X \in \mathcal{X}$ and any $\mathbf{w} \in \mathbb{Z}_q^{\bar{m}}$, the algorithm $\mathcal{H}.\mathrm{TrapEval}_{\mathcal{R}}(td, K, X)_{\mathcal{R}}$ is well-defined, and the statistical distance between $(\mathbf{A}, K', (\mathbf{R}'_X)^T\mathbf{w})$ and $(\mathbf{A}, K, \mathbf{R}_X^T\mathbf{w})$ is negligible in κ, where $(\mathbf{R}'_X, \mathbf{S}'_X) = \mathcal{H}.\mathrm{TrapEval}_{\mathcal{R}}(td, K', X)$, and $(\mathbf{R}_X, \mathbf{S}_X) = \mathcal{H}.\mathrm{TrapEval}_{\mathcal{R}}(td, K, X)$.*
2. *For any $(K', td) \leftarrow \mathcal{H}.\mathrm{TrapGen}_{\mathcal{R}}(1^\kappa, \mathbf{A}, \mathbf{B})$, any $X \in \mathcal{X}$, any uniformly random $\mathbf{v} \in \mathbb{Z}_q^{\bar{m}}$, and any uniformly random $\mathbf{u} \leftarrow \mathbb{Z}_q^m$, the statistical distance between $(\mathbf{A}, K', \mathbf{v}, (\mathbf{R}'_X)^T\mathbf{v})$ and $(\mathbf{A}, K', \mathbf{v}, \mathbf{u})$ is negligible in κ, where $(\mathbf{R}'_X, \mathbf{S}'_X) = \mathcal{H}.\mathrm{TrapEval}_{\mathcal{R}}(td, K', X)$.*

Remark 1. First, since $s_1(\mathbf{R}'_X) \leq \beta$ holds with overwhelming probability, we have that $\|(\mathbf{R}'_X)^T\mathbf{w}\| \leq \beta\|\mathbf{w}\|$. Thus, the first condition implicitly implies that $\|\mathbf{R}_X^T\mathbf{w}\| \leq \beta\|\mathbf{w}\|$ holds with overwhelming probability for any $K \leftarrow \mathcal{H}.\mathrm{Gen}(1^\kappa)$, $X \in \mathcal{X}$, and $(\mathbf{R}_X, \mathbf{S}_X) = \mathcal{H}.\mathrm{TrapEval}\ (td, K, X)$. Second, we note that the well-distributed hidden matrices property of PHF only holds when the information (except that is already leaked via the key K') of the trapdoor td is hidden. This means that it provides no guarantee when some information of $\mathbf{R}_X$ for any $X \in \mathcal{X}$ (which is usually related to the trapdoor td) is given public. However, for a PHF with high min-entropy, this property still holds when the information of $\mathbf{R}_X^T\mathbf{v}$ for a uniformly random vector $\mathbf{v}$ is leaked.

For appropriate choices of parameters, the work [2] implicitly showed that the Type-I PHF construction satisfied the high min-entropy property. Now, we show that our Type-II PHF construction with $\mathcal{R} = \mathbb{Z}$ also has the high min-entropy property. The setting for $\mathcal{R} = R$ may also be obtained for parameters such that the leftover hash lemma is applicable.

Theorem 7. *Let integers $\bar{n}, \bar{m}, q$ be some polynomials in the security parameter κ, and let $k = \lceil \log_2 q \rceil$. For any $\ell, v \in \mathbb{Z}$ and $L = 2^\ell$, let $N \leq 16v^2\ell, \eta \leq 4v\ell$ and $CF = \{CF_X\}_{X\in[L]}$ be defined as in Lemma 1. Then, for large enough $\bar{m} = O(\bar{n}\log q)$, the hash function $\mathcal{H} : [L] \rightarrow \mathcal{R}_q^{\bar{n}\times\bar{n}k}$ with $\mathcal{R} = \mathbb{Z}$ given in Definition 6 (and proved in Theorem 3) is a PHF with high min-entropy.*

Proof. For any $\mathbf{w} \in \mathbb{Z}_q^{\bar{m}}$, let $f_{\mathbf{w}} : \mathbb{Z}_q^{\bar{m}\times\bar{n}k} \rightarrow \mathbb{Z}_q^{\bar{n}k}$ be the function defined by $f_{\mathbf{w}}(\mathbf{X}) = \mathbf{X}^T\mathbf{w} \in \mathbb{Z}_q^{\bar{n}k}$. By the definition of $\mathcal{H}.\mathrm{TrapGen}_{\mathbb{Z}}$ in Theorem 3, for any $(K', td) \leftarrow \mathcal{H}.\mathrm{TrapGen}_{\mathbb{Z}}(1^\kappa, \mathbf{A}, \mathbf{G})$, we have $td = (\hat{\mathbf{R}}, \{\mathbf{R}_i\}_{i\in\{0,\ldots,\mu-1\}}, z^*)$. Denote $I = \{f_{\mathbf{w}}(\hat{\mathbf{R}}), \{f_{\mathbf{w}}(\mathbf{R}_i)\}_{i\in\{0,\ldots,\mu-1\}})\}$. First, it is easy to check that the algorithm $\mathcal{H}.\mathrm{TrapEval}_{\mathbb{Z}}(td, K, X)$ is well-defined for any $K \in \mathcal{K} = \mathbb{Z}_q^{\bar{n}\times\bar{n}k}$ and $X \in \mathcal{X}$. In addition, given $I = \{f_{\mathbf{w}}(\hat{\mathbf{R}}), \{f_{\mathbf{w}}(\mathbf{R}_i)\}_{i\in\{0,\ldots,\mu-1\}})\}$ and (K, X, z^*) as inputs, there exists a public algorithm that computes $\mathbf{R}_X^T\mathbf{w}$ by simulating the algorithm Tcomp in Theorem 3, where $(\mathbf{R}_X, \mathbf{S}_X) = \mathcal{H}.\mathrm{TrapEval}_{\mathbb{Z}}(td, K, X)$. To prove that $\mathcal{H}$ satisfies the first condition of high min-entropy, it suffices to show that K' is statistically close to uniform over $(\mathbb{Z}_q^{\bar{n}\times\bar{n}k})^{\mu+1}$ conditioned on I and z^* (recall that the real key K of $\mathcal{H}$ is uniformly distributed over $(\mathbb{Z}_q^{\bar{n}\times\bar{n}k})^{\mu+1}$ by Definition 6). Since each matrix in the key K' always has a form of $\mathbf{A}\tilde{\mathbf{R}} + b\mathbf{G}$ for some randomly chosen $\tilde{\mathbf{R}} \leftarrow (D_{\mathbb{Z}^{\bar{m}},s})^{\bar{n}k}$, and a bit $b \in \{0,1\}$ depending on a random $z^* \leftarrow [N]$. Using a standard hybrid argument, it is enough to show that conditioned on $\mathbf{A}$ and $f_{\mathbf{w}}(\tilde{\mathbf{R}})$, $\mathbf{A}\tilde{\mathbf{R}}$ is statistically close to uniform over $\mathbb{Z}_q^{\bar{n}\times\bar{n}k}$.

Let $f'_{\mathbf{w}} : \mathbb{Z}_q^{\bar{m}} \to \mathbb{Z}_q$ be defined by $f'_{\mathbf{w}}(\mathbf{x}) = \mathbf{x}^T\mathbf{w}$, and let $\tilde{\mathbf{R}} = (\mathbf{r}_1, \dots, \mathbf{r}_{\bar{n}k})$. Then, $f_{\mathbf{w}}(\tilde{\mathbf{R}}) = (f'_{\mathbf{w}}(\mathbf{r}_1), \dots, f'_{\mathbf{w}}(\mathbf{r}_{\bar{n}k}))^T \in \mathbb{Z}_q^{\bar{n}k}$. By Lemma 2, the guessing probability $\gamma(\mathbf{r}_i)$ is at most $2^{1-\bar{m}}$ for all $i \in \{1, \dots, \bar{n}k\}$. By the generalized leftover hash lemma in [23], conditioned on $\mathbf{A}$ and $f'_{\mathbf{w}}(\mathbf{r}_i) \in \mathbb{Z}_q$, the statistical distance between $\mathbf{A}\mathbf{r}_i \in \mathbb{Z}_q^{\bar{n}}$ and uniform over $\mathbb{Z}_q^{\bar{n}}$ is at most $\frac{1}{2} \cdot \sqrt{2^{1-\bar{m}} \cdot q^{\bar{n}} \cdot q}$, which is negligible if we set $\bar{m} = O(\bar{n}\log q) > (\bar{n}+1)\log q + \omega(\log \bar{n})$. Using a standard hybrid argument, we have that conditioned on $\mathbf{A}$ and $f_{\mathbf{w}}(\tilde{\mathbf{R}})$, the matrix $\mathbf{A}\tilde{\mathbf{R}} = (\mathbf{A}\mathbf{r}_1 \| \dots \| \mathbf{A}\mathbf{r}_{\bar{n}k})$ is statistically close to uniform over $\mathbb{Z}_q^{\bar{n}\times\bar{n}k}$.

Now, we show that $\mathcal{H}$ satisfies the second condition in Definition 9. By Theorem 3 for any input X and $(\mathbf{R}_X, \mathbf{S}_X) = \mathcal{H}.\mathrm{TrapEval}(td, K', X)$, we always have that $\mathbf{R}_X = \hat{\mathbf{R}} + \tilde{\mathbf{R}}$ for some $\tilde{\mathbf{R}}$ that is independent from $\hat{\mathbf{R}}$. Let $\mathbf{R}_X^T\mathbf{v} = \hat{\mathbf{R}}^T\mathbf{v} + \tilde{\mathbf{R}}^T\mathbf{v} = \hat{\mathbf{u}} + \tilde{\mathbf{u}}$, it suffices to show that given K' and $\mathbf{v}$, the element $\hat{\mathbf{u}} = \hat{\mathbf{R}}^T\mathbf{v}$ is uniformly random. Since $\hat{\mathbf{R}} \leftarrow (D_{\mathbb{Z}^{\bar{m}},s})^{\bar{n}k}$ for $s \geq \omega(\sqrt{\log \bar{m}})$ is only used to generate the matrix $\hat{\mathbf{A}} = \mathbf{A}\hat{\mathbf{R}} - (-1)^c \cdot \mathbf{G}$ in the key K', we have that for large enough $\bar{m} = O(\bar{n}\log q)$, the pair $(\mathbf{A}\hat{\mathbf{R}}, \hat{\mathbf{u}}^T = \mathbf{v}^T\hat{\mathbf{R}})$ is statistically close to uniform over $\mathbb{Z}_q^{\bar{n}\times\bar{n}k} \times \mathbb{Z}_q^{\bar{n}k}$ by the fact in Lemma 5.[8] Thus, $\mathbf{R}_X^T\mathbf{v} = \hat{\mathbf{R}}^T\mathbf{v} + \tilde{\mathbf{R}}^T\mathbf{v}$ is statistically close to uniform over $\mathbb{Z}_q^{\bar{n}k}$. This completes the proof 7. □

Remark 2. Our initial attempt to introduce the high min-entropy property in [60] is mainly for the generic construction of IBE from lattice-based PHFs. Although our improved IBE construction given in Sec. 6 does not need the high min-entropy in the security proof, we would like to still keep the definition of min-entropy property and we believe that it may be useful for other applications.

4 Short Signatures from Lattice-based PHFs

In this section, we first give a generic construction of signatures from PHFs, and then we give two improved signatures by using the concrete Type-II and Type-III PHF constructions, respectively.

4.1 A Generic Signature Scheme from Lattice-based PHFs

Let integers $\ell, n, m', q \in \mathbb{Z}, \beta \in \mathbb{R}$ be some polynomials in the security parameter κ, and let $k = \lceil \log_2 q \rceil$. Let ring $\mathcal{R}$ be either the integer ring $\mathbb{Z}$ or the polynomial

[8] This is because one can first construct a new uniformly random matrix $\mathbf{A}'$ by appending the row vector $\mathbf{v}^T$ to the rows of $\mathbf{A}$, and then apply the fact in Lemma 5.

ring $\mathbb{Z}[x]/(x^n+1)$ with n being a power of 2. Let $\mathcal{R}_q = \mathcal{R}/(q\mathcal{R})$ be the quotient ring. Let $n=1$ if $\mathcal{R} = \mathbb{Z}$. Let $\mathcal{H} = (\mathcal{H}.\mathrm{Gen}_R, \mathcal{H}.\mathrm{Eval}_R)$ be a PHF from $\{0,1\}^\ell$ to $\mathcal{R}_q^{\bar{n}\times m'}$. Let $\bar{m} = O(\bar{n}\log q)$, $m = \bar{m}+m'$, and large enough $s > \max(\beta, \sqrt{m}) \cdot \omega(\sqrt{\log n}) \in \mathbb{R}$ be the system parameters. Our generic signature scheme $\mathcal{SIG} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ is defined as follows.

$\mathsf{KeyGen}(1^\kappa)$**:** Given a security parameter κ as inputs, first compute $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}_{\mathcal{R}}(1^{\bar{n}}, 1^{\bar{m}}, q, \mathbf{I}_{\bar{n}})$ such that $\mathbf{A} \in \mathcal{R}_q^{\bar{n}\times\bar{m}}$, $\mathbf{R} = \mathcal{R}_q^{(\bar{m}-\bar{n}k)\times\bar{n}k}$, and randomly choose $\mathbf{u} \leftarrow \mathcal{R}_q^{\bar{n}}$. Then, compute $K \leftarrow \mathcal{H}.\mathrm{Gen}(1^\kappa)$, and return a pair of verification key and secret signing key $vk = (\mathbf{A}, \mathbf{u}, K)$ and $sk = (vk, \mathbf{R})$.

$\mathsf{Sign}(sk, M \in \{0,1\}^\ell)$**:** Given $sk = \mathbf{R}$ and a message M as inputs, compute $\mathbf{A}_M = (\mathbf{A} \| \mathrm{H}_K(M)) \in \mathcal{R}_q^{\bar{n}\times m}$, where $\mathrm{H}_K(M) = \mathcal{H}.\mathrm{Eval}(K, M) \in \mathcal{R}_q^{\bar{n}\times m'}$. Then, compute $\mathbf{e} \leftarrow \mathsf{SampleD}_{\mathcal{R}}(\mathbf{R}, \mathbf{A}_M, \mathbf{I}_{\bar{n}}, \mathbf{u}, s)$, and return $\sigma = \mathbf{e}$.

$\mathsf{Verify}(vk, M, \sigma)$**:** Given vk, a message M and a vector $\sigma = \mathbf{e}$ as inputs, compute $\mathbf{A}_M = (\mathbf{A} \| \mathrm{H}_K(M)) \in \mathcal{R}_q^{\bar{n}\times m}$, where $\mathrm{H}_K(M) = \mathcal{H}.\mathrm{Eval}(K, M) \in \mathcal{R}_q^{\bar{n}\times m'}$. Return 1 if $\|\mathbf{e}\| \leq s\sqrt{mn}$ and $\mathbf{A}_M\mathbf{e} = \mathbf{u}$, else return 0.

The correctness of our scheme $\mathcal{SIG}$ can be easily checked. Besides, the schemes with linear verification keys in [13,47] can be seen as instantiations of $\mathcal{SIG}$ with the Type-I PHF construction in Theorem 1.[9] Since the size of the verification key is mainly determined by the key size of $\mathcal{H}$, one can instantiate $\mathcal{H}$ with our efficient Type-II and Type-III PHF constructions. As for the security, we have the following theorem.

Theorem 8. *Let $\ell, \bar{n}, \bar{m}, m', q \in \mathbb{Z}$ and $\bar{\beta}, \beta, s \in \mathbb{R}$ be some polynomials in the security parameter κ, and let $m = \bar{m} + m'$. If $\mathcal{H} = (\mathcal{H}.\mathrm{Gen}_{\mathcal{R}}, \mathcal{H}.\mathrm{Eval}_{\mathcal{R}})$ be a $(1, \mathsf{poly}, \beta, \gamma, \delta)$-*PHF *from $\{0,1\}^\ell$ to $\mathcal{R}_q^{\bar{n}\times m'}$ with $\gamma = \mathrm{negl}(\kappa)$ and noticeable $\delta > 0$. Then, for large enough $\bar{m} = O(\bar{n}\log q)$ and $s > \max(\beta, \sqrt{m}) \cdot \omega(\sqrt{\log \bar{m}}) \in \mathbb{R}$, if there exists a PPT forger $\mathcal{F}$ breaking the* EUF-CMA *security of $\mathcal{SIG}$ with non-negligible probability $\epsilon > 0$, there exists an algorithm $\mathcal{B}$ solving the* $\mathrm{ISIS}_{q,\bar{m},\bar{\beta}}$ *over $\mathcal{R}$ problem for $\bar{\beta} = \beta s\sqrt{mn}\cdot\omega(\sqrt{\log \bar{m}})$ with probability at least $\epsilon' \geq \epsilon\delta - \mathrm{negl}(\kappa)$.*

*Moreover, if $\mathcal{H} = (\mathcal{H}.\mathrm{Gen}_{\mathcal{R}}, \mathcal{H}.\mathrm{Eval}_{\mathcal{R}})$ is a $(1, v, \beta, \gamma, \delta)$-*PHF *for some prior fixed polynomial $v = \mathsf{poly}(n)$, then the resulting signature scheme is secure against any PPT forger making at most $Q \leq v$ signing queries (namely, it satisfies the* EUF-qCMA *security).*

Since a proof sketch is given in Section 1.4, we omit the details of the proof. Let $\mathcal{SIG}_1$ denote the signature scheme obtained by instantiating $\mathcal{SIG}$ with our Type-II $(1, v, \beta)$-PHF construction with $\mathcal{R} = \mathbb{Z}$ in Definition 6. Then, the verification key of $\mathcal{SIG}_1$ has $O(\log n)$ matrices in $\mathbb{Z}_q^{n\times nk}$ and each signature of $\mathcal{SIG}_1$ consists of a single lattice vector.

Corollary 1. *Let $n, q \in \mathbb{Z}$ be polynomials in the security parameter κ. Let $\bar{m} = O(n\log q), v = \mathsf{poly}(n)$ and $\ell = n$. If there exists a PPT forger $\mathcal{F}$ breaking*

[9] Note that the scheme in [13] used a syndrome $\mathbf{u} = \mathbf{0}$, we prefer to use a random chosen syndrome $\mathbf{u} \leftarrow \mathbb{Z}_q^n$ as that in [47] for simplifying the security analysis.

the EUF-qCMA *security of $\mathcal{SIG}_1$ with non-negligible probability ϵ and making at most $Q \leq v$ signing queries, then there exists an algorithm $\mathcal{B}$ solving the* $\mathrm{ISIS}_{q,\bar{m},\bar{\beta}}$ *problem over $\mathcal{R} = \mathbb{Z}$ for $\bar{\beta} = v^2 \cdot \tilde{O}(n^{5.5})$ with probability at least* $\epsilon' \geq \frac{\epsilon}{16nv^2} - \mathrm{negl}(\kappa)$.

Similarly, let $\mathcal{SIG}_2$ denote the signature scheme obtained by instantiating $\mathcal{SIG}$ with our Type-III $(1, v, \beta)$-PHF construction with $\mathcal{R} = R$ in Definition 8. Then, the verification key of $\mathcal{SIG}_2$ has $O(1)$ matrices in $R_q^{1\times k}$ and each signature of $\mathcal{SIG}_2$ consists of a single ring vector.

Corollary 2. *Let $n, q \in \mathbb{Z}$ be polynomials in the security parameter κ. Let $\bar{m} = O(n \log q), v = \mathsf{poly}(n)$ and $\ell = n$. If there exists a PPT forger $\mathcal{F}$ breaking the* EUF-qCMA *security of $\mathcal{SIG}_2$ with non-negligible probability ϵ and making at most $Q \leq v$ signing queries, then there exists an algorithm $\mathcal{B}$ solving the* $\mathrm{ISIS}_{q,\bar{m},\bar{\beta}}$ *problem over $\mathcal{R} = R$ for $\bar{\beta} = v^2 \cdot \tilde{O}(n^{7.5+2/c})$ with probability at least* $\epsilon' \geq \frac{\epsilon}{16nv^2} - \mathrm{negl}(\kappa)$.

4.2 Improved Short Signature on General Lattices

Since both $\mathcal{SIG}_1$ and $\mathcal{SIG}_2$ have reduction loss about $16nv^2$, by the requirement that $v \geq Q$ for security proof, our improvement requires the $\mathrm{ISIS}_{q,\bar{m},\bar{\beta}}$ problem over $\mathcal{R}$ to be hard for $\bar{\beta} = Q^2 \cdot \tilde{O}(n^{5.5})$ for $\mathcal{SIG}_1$ or $\bar{\beta} = Q^2 \cdot \tilde{O}(n^{7.5+2/c})$ for $\mathcal{SIG}_2$, which means that the modulus q should be much bigger than Q^2. Even though q is still a polynomial of n in an asymptotic sense, it might be very large in practice. In this section, we give an improved signature scheme $\mathcal{SIG}_3$ from our Type-II PHF with $\mathcal{R} = \mathbb{Z}$, which removes the direct dependency on Q from $\bar{\beta}$ by introducing a short tag about $O(\log Q)$ bits to each signature. For example, this only increases about 30 bits to each signature for a number $Q = 2^{30}$ of the forger's signing queries.

At a high level, our basic idea is to relax the requirement on a $(1, v, \beta)$-PHF $\mathcal{H} = \{\mathrm{H}_K\}$ so that a much smaller $v = \omega(\log n)$ can be used by employing a simple weak PHF $\mathcal{H}' = \{\mathrm{H}'_{K'}\}$ (recall that $v \geq Q$ is required in the generic scheme $\mathcal{SIG}$). Concretely, for each message M to be signed, instead of using $\mathrm{H}_K(M)$ in the signing algorithm of $\mathcal{SIG}$, we choose a short random tag $\mathbf{t}$, and compute $\mathrm{H}'_{K'}(\mathbf{t}) + \mathrm{H}_K(M)$ to generate the signature on M. Thus, if the trapdoor keys of both PHFs are generated by using the same "generators" $\mathbf{A}$ and $\mathbf{G}$, we have that $\mathrm{H}'_{K'}(\mathbf{t}) + \mathrm{H}_K(M) = \mathbf{A}(\mathbf{R}'_{\mathbf{t}} + \mathbf{R}_M) + (\mathbf{S}'_{\mathbf{t}} + \mathbf{S}_M)\mathbf{G}$, where $\mathrm{H}'_{K'}(\mathbf{t}) = \mathbf{A}\mathbf{R}'_{\mathbf{t}} + \mathbf{S}'_{\mathbf{t}}\mathbf{G}$ and $\mathrm{H}_K(M) = \mathbf{A}\mathbf{R}_M + \mathbf{S}_M\mathbf{G}$. Moreover, if we can ensure that $\mathbf{S}'_{\mathbf{t}} + \mathbf{S}_M \in \mathcal{I}_n$ when $\mathbf{S}'_{\mathbf{t}} \in \mathcal{I}_n$ or $\mathbf{S}_M \in \mathcal{I}_n$, then $\mathbf{S}_M$ is not required to be invertible for all the Q signing messages. In particular, $v = \omega(\log n)$ can be used if the probability that $\mathbf{S}'_{\mathbf{t}} + \mathbf{S}_M \in \mathcal{I}_n$ is invertible for all the Q signing messages, and $\mathbf{S}'_{\mathbf{t}^*} + \mathbf{S}_{M^*} = \mathbf{0}$ for the forged signature on the pair $(\mathbf{t}^*, M^*)$, is noticeable.

Actually, the weak PHF $\mathcal{H}'$ and the $(1, v, \beta)$-PHF $\mathcal{H} = (\mathcal{H}.\mathrm{Gen}, \mathcal{H}.\mathrm{Eval})$ are, respectively, the first instantiated Type-I PHF $\mathcal{H}'$ with $\mathcal{R} = \mathbb{Z}$ in Theorem 2 and the Type-II PHF $\mathcal{H} = (\mathcal{H}.\mathrm{Gen}, \mathcal{H}.\mathrm{Eval})$ with $\mathcal{R} = \mathbb{Z}$ given in Definition 6. Since $\mathcal{H}'$ is very simple, we directly plug its construction into our signature scheme

$\mathcal{SIG}_2$. Specifically, let $n, q \in \mathbb{Z}$ be some polynomials in the security parameter κ, and let $k = \lceil \log_2 q \rceil, \bar{m} = O(n \log q), m = \bar{m} + nk$ and $s = \tilde{O}(n^{2.5}) \in \mathbb{R}$. Let $H : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n \times n}$ be the FRD encoding in [2] such that for any vector $\mathbf{v} = (v, 0 \ldots, 0)^T, \mathbf{v}_1, \mathbf{v}_2 \in \mathbb{Z}_q^n$, we have that $H(\mathbf{v}) = v\mathbf{I}_n$ and $H(\mathbf{v}_1) + H(\mathbf{v}_2) = H(\mathbf{v}_1 + \mathbf{v}_2)$ hold. For any $\mathbf{t} \in \{0,1\}^\ell$ with $\ell < n$, we naturally treat it as a vector in $\mathbb{Z}_q^n$ by appending it $(n - \ell)$ zero coordinates. The weak PHF $\mathcal{H}'$ from $\{0,1\}^\ell$ to $\mathbb{Z}_q^{n \times nk}$ has a form of $\mathrm{H}'_{K'}(\mathbf{t}) = \mathbf{A}_0 + H(\mathbf{t})\mathbf{G}$, where $K' = \mathbf{A}_0$. We restrict the domain of $\mathcal{H}'$ to be $\{0\} \times \{0,1\}^\ell$ for $\ell \leq n - 1$ such that $\mathbf{S}'_{\mathbf{t}} + \mathbf{S}_M$ is invertible when $(\mathbf{S}'_{\mathbf{t}}, \mathbf{S}_M) \neq (\mathbf{0}, \mathbf{0})$. Our signature scheme $\mathcal{SIG}_3 = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ is defined as follows.

$\mathsf{KeyGen}(1^\kappa)$**:** Given a security parameter κ as inputs, first compute $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^n, 1^{\bar{m}}, q, \mathbf{I}_n)$ such that $\mathbf{A} \in \mathbb{Z}_q^{n \times \bar{m}}$, $\mathbf{R} = \mathbb{Z}_q^{(\bar{m} - nk) \times nk}$. Randomly choose $\mathbf{A}_0 \leftarrow \mathbb{Z}_q^{n \times nk}, \mathbf{u} \leftarrow \mathbb{Z}_q^n$. Finally, compute $K \leftarrow \mathcal{H}.\mathrm{Gen}(1^\kappa)$, return $vk = (\mathbf{A}, \mathbf{A}_0, \mathbf{u}, K)$ and $sk = (vk, \mathbf{R})$.

$\mathsf{Sign}(sk, M \in \{0,1\}^n)$**:** Given the secret key sk and a message M as inputs, randomly choose $\mathbf{t} \leftarrow \{0,1\}^\ell$, and compute $\mathbf{A}_{M,\mathbf{t}} = (\mathbf{A} \| (\mathbf{A}_0 + H(0\|\mathbf{t})\mathbf{G}) + \mathrm{H}_K(M)) \in \mathbb{Z}_q^{n \times m}$, where $\mathrm{H}_K(M) = \mathcal{H}.\mathrm{Eval}(K, M) \in \mathbb{Z}_q^{n \times nk}$. Then, compute $\mathbf{e} \leftarrow \mathsf{SampleD}(\mathbf{R}, \mathbf{A}_{M,\mathbf{t}}, \mathbf{I}_n, \mathbf{u}, s)$, and return the signature $\sigma = (\mathbf{e}, \mathbf{t})$.

$\mathsf{Verify}(vk, M, \sigma)$**:** Given vk, message M and $\sigma = (\mathbf{e}, \mathbf{t})$ as inputs, compute $\mathbf{A}_{M,\mathbf{t}} = (\mathbf{A} \| (\mathbf{A}_0 + H(0\|\mathbf{t})\mathbf{G}) + \mathrm{H}_K(M)) \in \mathbb{Z}_q^{n \times m}$, where $\mathrm{H}_K(M) = \mathcal{H}.\mathrm{Eval}(K, M) \in \mathbb{Z}_q^{n \times nk}$. Return 1 if $\|\mathbf{e}\| \leq s\sqrt{m}$ and $\mathbf{A}_{M,\mathbf{t}}\mathbf{e} = \mathbf{u}$. Otherwise, return 0.

Since $\mathbf{R}$ is a $\mathbf{G}$-trapdoor of $\mathbf{A}$, by padding with zero rows it can be extended to a $\mathbf{G}$-trapdoor for $\mathbf{A}_{M,\mathbf{t}}$ with the same quality $s_1(\mathbf{R}) \leq \sqrt{m} \cdot \omega(\sqrt{\log n})$. Since $s = \tilde{O}(n^{2.5}) > s_1(\mathbf{R}) \cdot \omega(\sqrt{\log n})$, the vector $\mathbf{e}$ output by $\mathsf{SampleD}$ follows the distribution $D_{\mathbb{Z}^m, s}$ satisfying $\mathbf{A}_{M,\mathbf{t}}\mathbf{e} = \mathbf{u}$. In other words, $\|\mathbf{e}\| \leq s\sqrt{m}$ holds with overwhelming probability by Lemma 2. This shows that $\mathcal{SIG}_3$ is correct.

Note that if we set $v = \omega(\log n)$, the key K only has $\mu = O(\log n)$ number of matrices in $\mathbb{Z}_q^{n \times nk}$ and each signature consists of a vector plus a short ℓ-bit tag. We have the following theorem for security.

Theorem 9. *Let $\ell, \bar{m}, n, q, v, Q \in \mathbb{Z}$ be polynomials in the security parameter κ. Let $\ell = O(\log n)$ depending on Q and $v = \omega(\log n)$, if there exists a PPT forger $\mathcal{F}$ breaking the* EUF-qCMA *security of $\mathcal{SIG}_3$ with non-negligible probability ϵ and making at most Q signing queries, there exists an algorithm $\mathcal{B}$ solving the* $\mathrm{ISIS}_{q,\bar{m},\bar{\beta}}$ *problem for $\bar{\beta} = \tilde{O}(n^{5.5})$ with probability at least $\epsilon' \geq \frac{\epsilon}{16 \cdot 2^\ell n v^2} - \mathrm{negl}(\kappa) = \frac{\epsilon}{Q \cdot \tilde{O}(n)}$.*

Proof. We now give the construction of algorithm $\mathcal{B}$, which simulates the attack environment for $\mathcal{F}$, and solves the $\mathrm{ISIS}_{q,\bar{m},\bar{\beta}}$ problem with probability at least $\frac{\epsilon}{Q \cdot \tilde{O}(n)}$. Formally, $\mathcal{B}$ first randomly chooses a vector $\mathbf{t}' \leftarrow \{0,1\}^\ell$ and hopes that $\mathcal{F}$ will output a forged signature with tag $\mathbf{t}^* = \mathbf{t}'$. Then, $\mathcal{B}$ simulates the EUF-qCMA game as follows:

KeyGen. Given an $\mathrm{ISIS}_{q,\bar{m},\bar{\beta}}$ challenge instance $(\mathbf{A},\mathbf{u}) \in \mathbb{Z}_q^{n\times\bar{m}} \times \mathbb{Z}_q^n$, the algorithm $\mathcal{B}$ first randomly chooses $\mathbf{R}_0 \leftarrow (D_{\mathbb{Z}^{\bar{m}},\omega(\sqrt{\log n})})^{nk}$, and computes $\mathbf{A}_0 = \mathbf{A}\mathbf{R}_0 - H(0\|\mathbf{t}')\mathbf{G}$. Then, compute $(K', td) \leftarrow \mathcal{H}.\mathrm{TrapGen}(1^\kappa, \mathbf{A}, \mathbf{G})$ as in Theorem 3. Finally, set $vk = (\mathbf{A}, \mathbf{A}_0, \mathbf{u}, K')$ and keep $(\mathbf{R}_0, td)$ private.

Signing. Given a message M, the algorithm $\mathcal{B}$ first randomly chooses a tag $\mathbf{t} \leftarrow \{0,1\}^\ell$. If $\mathbf{t}$ has been used in answering the signatures for more than v messages, $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ computes $(\mathbf{R}_M, \mathbf{S}_M) = \mathcal{H}.\mathrm{TrapEval}(td, K', M)$ as in Theorem 3. Then, we have $\mathbf{A}_{M,\mathbf{t}} = (\mathbf{A}\|(\mathbf{A}_0 + H(0\|\mathbf{t})\mathbf{G}) + \mathrm{H}_{K'}(M)) = (\mathbf{A}\|\mathbf{A}(\mathbf{R}_0+\mathbf{R}_M)+(H(0\|\mathbf{t})-H(0\|\mathbf{t}')+\mathbf{S}_M)\mathbf{G})$. Since $\mathbf{S}_M = b\mathbf{I}_n = H(b\|0)$ for some $b \in \{-1,0,1\}$, we have that $\hat{\mathbf{S}} = H(0\|\mathbf{t})-H(0\|\mathbf{t}')+\mathbf{S}_M = H(b\|(\mathbf{t}-\mathbf{t}'))$ holds by the homomorphic property of the FRD encoding H in [2]. $\mathcal{B}$ distinguishes the following two cases:

- $\mathbf{t} \neq \mathbf{t}'$ or $(\mathbf{t} = \mathbf{t}' \wedge b \neq 0)$: In both cases, we have that $\hat{\mathbf{S}}$ is invertible. In other words, $\hat{\mathbf{R}} = \mathbf{R}_0 + \mathbf{R}_M$ is a $\mathbf{G}$-trapdoor for $\mathbf{A}_{M,\mathbf{t}}$. Since $s_1(\mathbf{R}_0) \leq \sqrt{m}\cdot\omega(\sqrt{\log n})$ by Lemma 3 and $s_1(\mathbf{R}_M) \leq \tilde{O}(n^{2.5})$, we have $s_1(\hat{\mathbf{R}}) \leq \tilde{O}(n^{2.5})$. Then, compute $\mathbf{e} \leftarrow \mathsf{SampleD}(\hat{\mathbf{R}}, \mathbf{A}_{M,\mathbf{t}}, \hat{\mathbf{S}}, \mathbf{u}, s)$, and return the signature $\sigma = (\mathbf{e}, \mathbf{t})$. If we set an appropriate $s = \tilde{O}(n^{2.5}) \geq s_1(\hat{\mathbf{R}})\cdot\omega(\sqrt{\log n})$, then $\mathcal{B}$ can generate a valid signature on M with overwhelming probability by Proposition 1.
- $\mathbf{t} = \mathbf{t}' \wedge b = 0$: $\mathcal{B}$ aborts.

Forge. After making at most Q signing queries, $\mathcal{F}$ outputs a forged signature $\sigma^* = (\mathbf{e}^*, \mathbf{t}^*)$ on message $M^* \in \{0,1\}^n$ such that $\|\mathbf{e}^*\| \leq s\sqrt{m}$ and $\mathbf{A}_{M^*,\mathbf{t}^*}\mathbf{e}^* = \mathbf{u}$, where $\mathbf{A}_{M^*,\mathbf{t}^*} = (\mathbf{A}\|(\mathbf{A}_0 + H(0\|\mathbf{t}^*)\mathbf{G}) + \mathrm{H}_{K'}(M^*)) \in \mathbb{Z}_q^{n\times m}$. The algorithm $\mathcal{B}$ computes $(\mathbf{R}_{M^*}, \mathbf{S}_{M^*}) = \mathcal{H}.\mathrm{TrapEval}(td, K', M^*)$, and aborts the simulation if $\mathbf{t}^* \neq \mathbf{t}'$ or $\mathbf{S}_{M^*} \neq \mathbf{0}$. Else, we have $\mathbf{A}_{M^*,\mathbf{t}^*} = (\mathbf{A}\|\mathbf{A}(\mathbf{R}_0 + \mathbf{R}_{M^*})) = (\mathbf{A}\|\mathbf{A}\hat{\mathbf{R}})$, where $\hat{\mathbf{R}} = \mathbf{R}_0 + \mathbf{R}_{M^*}$. Finally, $\mathcal{B}$ outputs $\hat{\mathbf{e}} = (\mathbf{I}_{\bar{m}}\|\hat{\mathbf{R}})\mathbf{e}^*$ as its own solution.

By the definition of the $\mathrm{ISIS}_{q,\bar{m},\bar{\beta}}$ problem, $(\mathbf{A},\mathbf{u})$ is uniformly distributed over $\mathbb{Z}_q^{n\times\bar{m}} \times \mathbb{Z}_q^n$. Since $\mathbf{R}_0 \leftarrow (D_{\mathbb{Z}^{\bar{m}},\omega(\sqrt{\log n})})^{nk}$, we have that $\mathbf{A}_0 \in \mathbb{Z}_q^{n\times nk}$ is statistically close to uniform over $\mathbb{Z}_q^{n\times nk}$ by Lemma 5. In addition, by Theorem 3 the simulated key K' is statistically close to the real key K. Thus, the distribution of the simulated verification key vk is statistically close to that of the real one.

Let $M_1, \ldots, M_u$ be all the messages in answering the signing queries that $\mathcal{B}$ happens to use the same tag $\mathbf{t} = \mathbf{t}'$, and let $(\mathbf{R}_{M_i}, \mathbf{S}_{M_i}) = \mathcal{H}.\mathrm{TrapEval}(td, K', M_i)$ for $i \in \{1,\ldots,u\}$. Then, the algorithm $\mathcal{B}$ will abort in the simulation if and only if either of the following two conditions hold:

- Some tag $\mathbf{t}$ is used in answering the signatures for more than v messages,
- $\mathbf{S}_{M_i}$ is not invertible for some $i \in \{1,\ldots,u\}$, or $\mathbf{S}_{M^*} \neq \mathbf{0}$, or $\mathbf{t}^* \neq \mathbf{t}'$.

Since the forger $\mathcal{F}$ will make at most $Q = \mathsf{poly}(\kappa)$ signing queries, we can choose $\ell = O(\log n)$ such that $\frac{Q}{2^\ell} \leq \frac{1}{2}$. Note that $\mathcal{B}$ always randomly chooses a tag $\mathbf{t} \leftarrow \{0,1\}^\ell$ for each signing message, the probability that $\mathcal{B}$ uses any tag $\mathbf{t}$ in answering the signatures for more than v messages is less than $Q^2\cdot(\frac{Q}{2^\ell})^v$ by a similar analysis in [35], which is negligible by our setting of $v = \omega(\log n)$.

In particular, the probability that $\mathcal{B}$ will use the same tag $\mathbf{t} = \mathbf{t}'$ in answering the signatures for $u \geq v$ messages is also negligible. Conditioned on $u \leq v$, the probability that $\mathbf{S}_{M_i}$ is invertible for all $i \in \{1, \dots, u\}$ and $\mathbf{S}_{M^*} = \mathbf{0}$ (using the fact that $M^* \notin \{M_1, \dots, M_u\}$) is at least $\delta = \frac{1}{16nv^2} - \mathrm{negl}(\kappa)$ by Theorem 3. Note that $\mathbf{t}'$ is randomly chosen and is statistically hidden from $\mathcal{F}$, the probability $\Pr[\mathbf{t}^* = \mathbf{t}']$ is at least $\frac{1}{2^\ell} - \mathrm{negl}(\kappa)$. Thus, if the forger $\mathcal{F}$ can attack the EUF-qCMA security of $\mathcal{SIG}_3$ with probability ϵ in the real game, then it will also output a valid forgery $(M^*, \mathbf{e}^*)$ in the simulated game with probability at least $(\epsilon - Q^2(\frac{Q}{2^\ell})^v) \cdot \delta \cdot (\frac{1}{2^\ell} - \mathrm{negl}(\kappa)) = \frac{\epsilon}{2^\ell \cdot 16nv^2} - \mathrm{negl}(\kappa) = \frac{\epsilon}{Q \cdot \tilde{O}(n)}$ (note that $\mathcal{F}$'s success probability ϵ might be correlated with the first abort condition).

Now, we show that $\hat{\mathbf{e}} = (\mathbf{I}_{\bar{m}} \| \hat{\mathbf{R}})\mathbf{e}^*$ is a valid solution to the $\mathrm{ISIS}_{q,\bar{m},\bar{\beta}}$ instance $(\mathbf{A}, \mathbf{u})$. By the conditions in the verification algorithm, we have that $\mathbf{A}_{M^*,\mathbf{t}^*}\mathbf{e}^* = \mathbf{u}$ and $\|\mathbf{e}^*\| \leq s\sqrt{m}$. Since $s_1(\mathbf{R}_0) \leq \sqrt{m} \cdot \omega(\sqrt{\log n})$ by Lemma 3 and $s_1(\mathbf{R}_{M^*}) \leq \beta = \tilde{O}(n^{2.5})$ by Theorem 3, we have that $\|\hat{\mathbf{e}}\| \leq \tilde{O}(n^{2.5}) \cdot s\sqrt{m} = \tilde{O}(n^{5.5}) = \bar{\beta}$. This finally completes the proof. □

5 Fully-secure Short Signatures from PHFs

The three concrete signature schemes (i.e., $\mathcal{SIG}_1, \mathcal{SIG}_2$ and $\mathcal{SIG}_3$) given in the above section can only achieve a weak EUF-qCMA security because we have to set the scheme parameter depending on the number Q of the adversary's signing queries. For $\mathcal{SIG}_3$, this dependence is somewhat loose, because there is only one parameter $\ell = O(\log n)$ (i.e., the tag length) that directly depends on Q due to the requirement $Q/2^\ell < 1/2$ in the security proof, and $\mathcal{SIG}_3$ can be easily made to achieve the full EUF-CMA security with a choice of $\ell = \omega(\log n)$ such that $Q/2^\ell = \mathrm{negl}(n) < 1/2$ holds for any polynomially-bounded Q. The main problem is that the resulting security proof becomes less interesting due to the super-polynomial factor $2^{\omega(\log n)}$ in the reduction loss.

In this section, we give two new signature schemes from the Type-II and Type-III PHFs, which have similar structures to $\mathcal{SIG}_3$ but directly achieve full EUF-CMA security with asymptotically the same key sizes and reduction loss. Technically, we will replace the simple weak PHF component in $\mathcal{SIG}_3$ with a set of weak PHFs to realize the confined guessing technique [9] such that we can somehow dynamically set an appropriate "tag length" after knowing the number Q of the signing queries in the security proof.

5.1 A Fully-secure Short Signature on General Lattices

Let $\ell = \omega(\log n) < n$ be a function depending on the security parameter κ, and let $d = \lceil \log \ell \rceil$. For any vector $\mathbf{t} \in \{0,1\}^\ell$, let $\mathbf{t}_j \in \{0,1\}^j$ be the binary vector consisting of the first j bits of $\mathbf{t}$ (which means that $\mathbf{t}_\ell = \mathbf{t}$). Let $H : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n \times n}$ be the FRD encoding in [2] such that for any vector $\mathbf{v} = (v, 0 \dots, 0)^T, \mathbf{v}_1, \mathbf{v}_2 \in \mathbb{Z}_q^n$, we have that $H(\mathbf{v}) = v\mathbf{I}_n$ and $H(\mathbf{v}_1) + H(\mathbf{v}_2) = H(\mathbf{v}_1 + \mathbf{v}_2)$ hold. For any $\mathbf{t} \in \{0,1\}^{\ell'}$ with $\ell' < n$, we naturally treat it as a binary vector in $\mathbb{Z}_q^n$ by appending it $(n-\ell')$ zero coordinates. Our signature scheme $\mathcal{SIG}_4 = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ is defined as follows.

$\mathsf{KeyGen}(1^\kappa)$**:** Given a security parameter κ as input, first compute $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^n, 1^{\bar{m}}, q, \mathbf{I}_n)$ such that $\mathbf{A} \in \mathbb{Z}_q^{n\times \bar{m}}$, $\mathbf{R} = \mathbb{Z}_q^{(\bar{m}-nk)\times nk}$. Randomly choose $\mathbf{A}_i \leftarrow \mathbb{Z}_q^{n\times nk}$ for $i \in \{0, 1, \ldots, d\}$, and $\mathbf{u} \leftarrow \mathbb{Z}_q^n$. Finally, compute $K \leftarrow \mathcal{H}.\mathrm{Gen}(1^\kappa)$, return $vk = (\mathbf{A}, \{\mathbf{A}_i\}_{0\leq i\leq d}, \mathbf{u}, K)$ and $sk = (vk, \mathbf{R})$.

$\mathsf{Sign}(sk, M \in \{0,1\}^n)$**:** Given the secret key sk and a message M as inputs, randomly choose a tag $\mathbf{t} \leftarrow \{0,1\}^\ell$, let $\mathbf{t}_j \in \{0,1\}^j$ be the binary vector consisting of the first j bits of $\mathbf{t}$, where $1 \leq j \leq \ell$. Then, compute $\hat{\mathbf{A}}_j = \mathsf{Comp}_\ell(\{\mathbf{A}_i\}_{0\leq i\leq d-1}, j)$ for all $1 \leq j \leq \ell$ as shown in Fig. 1, and

$$\mathbf{A}_{M,\mathbf{t}} = (\mathbf{A} \| (\mathbf{A}_d + \sum_{j=1}^{\ell} \hat{\mathbf{A}}_j \cdot \mathbf{G}^{-1}(H(0\|\mathbf{t}_j)\mathbf{G}) + \mathrm{H}_K(M)) \in \mathbb{Z}_q^{n\times m},$$

where $\mathrm{H}_K(M) = \mathcal{H}.\mathrm{Eval}(K, M) \in \mathbb{Z}_q^{n\times nk}$. Finally, compute and return the signature $\sigma = (\mathbf{e}, \mathbf{t})$, where $\mathbf{e} \leftarrow \mathsf{SampleD}(\mathbf{R}, \mathbf{A}_{M,\mathbf{t}}, \mathbf{I}_n, \mathbf{u}, s)$.

$\mathsf{Verify}(vk, M, \sigma)$**:** Given vk, message M and $\sigma = (\mathbf{e}, \mathbf{t})$ as inputs, compute $\hat{\mathbf{A}}_j = \mathsf{Comp}_\ell(\{\mathbf{A}_i\}_{0\leq i\leq d-1}, j)$ for all $1 \leq j \leq \ell$ as shown in Fig. 1, and

$$\mathbf{A}_{M,\mathbf{t}} = (\mathbf{A} \| (\mathbf{A}_d + \sum_{j=1}^{\ell} \hat{\mathbf{A}}_j \cdot \mathbf{G}^{-1}(H(0\|\mathbf{t}_j)\mathbf{G}) + \mathrm{H}_K(M)) \in \mathbb{Z}_q^{n\times m},$$

where $\mathrm{H}_K(M) = \mathcal{H}.\mathrm{Eval}(K, M) \in \mathbb{Z}_q^{n\times nk}$. Return 1 if $\|\mathbf{e}\| \leq s\sqrt{m}$ and $\mathbf{A}_{M,\mathbf{t}}\mathbf{e} = \mathbf{u}$. Otherwise, return 0.

Since $\mathbf{R}$ is a $\mathbf{G}$-trapdoor of $\mathbf{A}$, by padding with zero rows it can be extended to a $\mathbf{G}$-trapdoor for $\mathbf{A}_{M,\mathbf{t}}$ with the same quality $s_1(\mathbf{R}) \leq \sqrt{\bar{m}} \cdot \omega(\sqrt{\log n})$. Since $s = \tilde{O}(n^{2.5}) > s_1(\mathbf{R}) \cdot \omega(\sqrt{\log n})$, the vector $\mathbf{e}$ output by $\mathsf{SampleD}$ follows the distribution $D_{\mathbb{Z}^m,s}$ satisfying $\mathbf{A}_{M,\mathbf{t}}\mathbf{e} = \mathbf{u}$. In other words, $\|\mathbf{e}\| \leq s\sqrt{m}$ holds with overwhelming probability by Lemma 2. This shows that $\mathcal{SIG}_4$ is correct.

Note that if we set $v = \omega(\log n)$, the key K only has $\mu = O(\log n)$ number of matrices in $\mathbb{Z}_q^{n\times nk}$ and each signature consists of a vector plus a short ℓ-bit tag. This means that the total number of matrix in the verification key is at most $d + 2 + \mu = O(\log n)$. We have the following theorem for security.

Theorem 10. *Let $\ell, \bar{m}, n, q, v \in \mathbb{Z}$ be polynomials in the security parameter κ. Let $\ell = \omega(\log n), v = \omega(\log n), d = \lceil \log \ell \rceil$. Then, if there exists a PPT forger $\mathcal{F}$ breaking the* EUF-CMA *security of $\mathcal{SIG}_4$ with non-negligible probability ϵ and making at most $Q = \mathsf{poly}(n)$ signing queries, there exists an algorithm $\mathcal{B}$ solving the* $\mathrm{ISIS}_{q,\bar{m},\bar{\beta}}$ *problem with probability at least $\epsilon' \geq \frac{\epsilon}{64\cdot Q\cdot nv^2} - \mathrm{negl}(\kappa)$ for some $\bar{\beta} = \tilde{O}(n^{5.5})$.*

Proof. We now give the construction of algorithm $\mathcal{B}$, which simulates the attack environment for $\mathcal{F}$, and solves the $\mathrm{ISIS}_{q,\bar{m},\bar{\beta}}$ problem with probability ϵ'. Since the number Q of the adversary's signing queries is bounded by a polynomial, there must exist an index $1 \le j^* \le \ell = \omega(\log n)$ such that $\frac{Q}{2^{j^*}} \le 1/2 < \frac{Q}{2^{j^*-1}}$. The algorithm $\mathcal{B}$ first randomly chooses a vector $\mathbf{t}' \leftarrow \{0,1\}^{j^*}$, and hopes that $\mathcal{F}$ will output a forged signature with a tag $\mathbf{t}^*$ whose first j^* bits are extactly $\mathbf{t}'$ (namely, $\mathbf{t}^*_{j^*} = \mathbf{t}'$). Then, $\mathcal{B}$ simulates the EUF-CMA game as follows:

KeyGen. Given an $\mathrm{ISIS}_{q,\bar{m},\bar{\beta}}$ challenge instance $(\mathbf{A}, \mathbf{u}) \in \mathbb{Z}_q^{n\times\bar{m}} \times \mathbb{Z}_q^n$, the algorithm $\mathcal{B}$ first randomly chooses $\mathbf{R}_i \leftarrow (D_{\mathbb{Z}^{\bar{m}},\omega(\sqrt{\log n})})^{nk}$ for $0 \le i \le d$. Then, let $(b_0^*, \dots, b_{d-1}^*) = \mathsf{BitDecomp}_\kappa(j^*)$, and compute $\mathbf{A}_i = \mathbf{A}\mathbf{R}_i + (1 - b_i^*)\mathbf{G}$ for $0 \le i \le d-1$ and $\mathbf{A}_d = \mathbf{A}\mathbf{R}_d - (-1)^c \cdot H(0\|\mathbf{t}')\mathbf{G}$, where c is the number of 1's in the binary vector $(b_0^*, \dots, b_{d-1}^*)$. Next, compute $(K', td) \leftarrow \mathcal{H}.\mathrm{TrapGen}(1^\kappa, \mathbf{A}, \mathbf{G})$ as in Theorem 3. Finally, set $vk = (\mathbf{A}, \{\mathbf{A}_i\}_{0\le i\le d}, \mathbf{u}, K')$ and keep $(\{\mathbf{R}_i\}_{0\le i\le d}, td)$ private.

Signing. Given a message M, the algorithm $\mathcal{B}$ first randomly chooses a tag $\mathbf{t} \leftarrow \{0,1\}^\ell$. Let $\mathbf{t}_j \in \{0,1\}^j$ be the binary vector consisting of the first j bits of $\mathbf{t}$, where $1 \le j \le \ell$. If $\mathbf{t}_{j^*}$ has been used in generating the signatures for more than v messages, $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ computes $\hat{\mathbf{A}}_j = \mathsf{Comp}_\ell(\{\mathbf{A}_i\}_{0\le i\le d-1}, j)$ for all $1 \le j \le \ell$ as shown in Fig. 1, $(\mathbf{R}_M, \mathbf{S}_M) = \mathcal{H}.\mathrm{TrapEval}(td, K', M)$, and

$$\mathbf{A}_{M,\mathbf{t}} = (\mathbf{A}\|(\mathbf{A}_d + \sum_{j=1}^{\ell} \hat{\mathbf{A}}_j \cdot \mathbf{G}^{-1}(H(0\|\mathbf{t}_j)\mathbf{G}) + \mathrm{H}_{K'}(M)).$$

Let $(\hat{\mathbf{R}}_j, \hat{\mathbf{S}}_j) = \mathsf{Tcomp}_\ell(\{\mathbf{A}_i, \mathbf{R}_i\}_{0\le i\le d-1}, j)$ for all $1 \le j \le \ell$, by the proof of Theorem 3, we have that $\hat{\mathbf{A}}_j = \mathbf{A}\hat{\mathbf{R}}_j + \hat{\mathbf{S}}_j$. In particular, we have that $s_1(\hat{\mathbf{R}}_j) \le d\bar{m}^{1.5} \cdot \omega(\sqrt{\log \bar{m}})$ and that $\hat{\mathbf{S}}_j = (-1)^c \cdot \mathbf{I}_n$ for $j = j^*$, and $\hat{\mathbf{S}}_j = \mathbf{0}$ otherwise. This means that we have $\mathbf{A}_{M,\mathbf{t}} = (\mathbf{A}\|\mathbf{A}\hat{\mathbf{R}} + \hat{\mathbf{S}}\mathbf{G})$, where

$$\begin{aligned}\hat{\mathbf{R}} &= \mathbf{R}_d + \textstyle\sum_{j=1}^{\ell} \hat{\mathbf{R}}_j \cdot \mathbf{G}^{-1}(H(0\|\mathbf{t}_j)\mathbf{G}) + \mathbf{R}_M,\\ \hat{\mathbf{S}} &= -(-1)^c \cdot H(0\|\mathbf{t}') + (-1)^c \cdot H(0\|\mathbf{t}_{j^*}) + \mathbf{S}_M.\end{aligned}$$

Since $\mathbf{S}_M = b\mathbf{I}_n = H(b\|0)$ for some $b \in \{-1,0,1\}$, we have that $\hat{\mathbf{S}} = H(b\|(-1)^c \cdot (\mathbf{t}_{j^*} - \mathbf{t}'))$ holds by the homomorphic property of the FRD encoding H in [2]. $\mathcal{B}$ distinguishes the following two cases:

- $\mathbf{t}_{j^*} \ne \mathbf{t}'$ or $(\mathbf{t}_{j^*} = \mathbf{t}' \wedge b \ne 0)$: In both cases, we have that $\hat{\mathbf{S}}$ is invertible. In other words, $\hat{\mathbf{R}}$ is a $\mathbf{G}$-trapdoor for $\mathbf{A}_{M,\mathbf{t}}$. Since $s_1(\mathbf{R}_i) \le \sqrt{\bar{m}}\cdot\omega(\sqrt{\log n})$ by Lemma 3 and $s_1(\mathbf{R}_M) \le \tilde{O}(n^{2.5})$, we have $s_1(\hat{\mathbf{R}}) \le \tilde{O}(n^{2.5})$. Then, compute $\mathbf{e} \leftarrow \mathsf{SampleD}(\hat{\mathbf{R}}, \mathbf{A}_{M,\mathbf{t}}, \hat{\mathbf{S}}, \mathbf{u}, s)$, and return the signature $\sigma = (\mathbf{e}, \mathbf{t})$. If we set an appropriate $s = \tilde{O}(n^{2.5}) \ge s_1(\hat{\mathbf{R}}) \cdot \omega(\sqrt{\log n})$, then $\mathcal{B}$ can generate a valid signature on M with overwhelming probability by Proposition 1.
- $\mathbf{t}_{j^*} = \mathbf{t}' \wedge b = 0$: $\mathcal{B}$ aborts.

Forge. After making at most Q signing queries, $\mathcal{F}$ outputs a forged signature $\sigma^* = (\mathbf{e}^*, \mathbf{t}^*)$ on message $M^* \in \{0,1\}^n$ such that $\|\mathbf{e}^*\| \le s\sqrt{m}$ and $\mathbf{A}_{M^*,\mathbf{t}^*}\mathbf{e}^* = \mathbf{u}$, where

$$\mathbf{A}_{M^*,\mathbf{t}^*} = (\mathbf{A}\|(\mathbf{A}_d + \sum_{j=1}^{\ell} \hat{\mathbf{A}}_j \cdot \mathbf{G}^{-1}(H(0\|\mathbf{t}_j^*)\mathbf{G}) + \mathrm{H}_{K'}(M^*)) \in \mathbb{Z}_q^{n\times m}.$$

The algorithm $\mathcal{B}$ computes $(\mathbf{R}_{M^*}, \mathbf{S}_{M^*}) = \mathcal{H}.\mathrm{TrapEval}(td, K', M^*)$, and aborts the simulation if $\mathbf{t}_{j^*}^* \ne \mathbf{t}'$ or $\mathbf{S}_{M^*} \ne \mathbf{0}$. Else, we have

$$\mathbf{A}_{M^*,\mathbf{t}^*} = (\mathbf{A}\|\mathbf{A}(\mathbf{R}_d + \sum_{j=1}^{\ell} \hat{\mathbf{R}}_j \cdot \mathbf{G}^{-1}(H(0\|\mathbf{t}_j^*)\mathbf{G}) + \mathbf{R}_{M^*})) = (\mathbf{A}\|\mathbf{A}\hat{\mathbf{R}}^*),$$

where $\hat{\mathbf{R}}^* = \mathbf{R}_0 + \sum_{j=1}^{\ell} \hat{\mathbf{R}}_j \cdot \mathbf{G}^{-1}(H(0\|\mathbf{t}_j^*)\mathbf{G}) + \mathbf{R}_{M^*}$. Finally, $\mathcal{B}$ outputs $\hat{\mathbf{e}} = (\mathbf{I}_{\bar{m}}\|\hat{\mathbf{R}}^*)\mathbf{e}^*$ as its own solution.

By the definition of the $\mathrm{ISIS}_{q,\bar{m},\bar{\beta}}$ problem, $(\mathbf{A}, \mathbf{u})$ is uniformly distributed over $\mathbb{Z}_q^{n\times\bar{m}} \times \mathbb{Z}_q^n$. Since $\mathbf{R}_i \leftarrow (D_{\mathbb{Z}^{\bar{m}},\omega(\sqrt{\log n})})^{nk}$, we have that $\mathbf{A}_i \in \mathbb{Z}_q^{n\times nk}$ is statistically close to uniform over $\mathbb{Z}_q^{n\times nk}$ by Lemma 5, where $0 \le i \le d$. In addition, by Theorem 3 the simulated PHF key K' is statistically close to the real key K. Thus, the distribution of the simulated verification key vk is statistically close to that of the real one.

Let $M_1, \dots, M_u$ be all the messages in the signing queries that $\mathcal{B}$ happens to use $\mathbf{t}' \in \{0,1\}^{j^*}$ in generating the signatures, and let $(\mathbf{R}_{M_i}, \mathbf{S}_{M_i}) = \mathcal{H}.\mathrm{TrapEval}(td, K', M_i)$ for $i \in \{1, \dots, u\}$. Then, $\mathcal{B}$ will abort in the simulation if and only if either of the following two conditions hold:

- Some $\mathbf{t}_{j^*} \in \{0,1\}^{j^*}$ has been used in generating the signatures for more than v messages;
- $\mathbf{S}_{M_i}$ is not invertible for some $i \in \{1, \dots, u\}$, or $\mathbf{S}_{M^*} \ne \mathbf{0}$, or $\mathbf{t}_{j^*}^* \ne \mathbf{t}'$.

Note that $\mathcal{B}$ always randomly chooses a tag $\mathbf{t} \leftarrow \{0,1\}^{\ell}$ for each signing message, the probability that the same $\mathbf{t}_{j^*} \in \{0,1\}^{j^*}$ is used in generating the signatures for more than v messages is less than $Q^2 \cdot (\frac{Q}{2^{j^*}})^v$ by a similar analysis in [35], which is negligible by our setting of $\frac{Q}{2^{j^*}} \le 1/2$ and $v = \omega(\log n)$. In particular, the probability that $\mathcal{B}$ will use $\mathbf{t}' \in \{0,1\}^{j^*}$ in generating the signatures for $u \ge v$ messages is also negligible. Conditioned on $u \le v$, the probability that $\mathbf{S}_{M_i}$ is invertible for all $i \in \{1, \dots, u\}$ and $\mathbf{S}_{M^*} = \mathbf{0}$ (using the fact that $M^* \notin \{M_1, \dots, M_u\}$) is at least $\delta = \frac{1}{16nv^2} - \mathrm{negl}(\kappa)$ by Theorem 3. Note that $\mathbf{t}'$ is randomly chosen and is statistically hidden from $\mathcal{F}$, we have that the probability $\Pr[\mathbf{t}_{j^*}^* = \mathbf{t}']$ is at least $\frac{1}{2^{j^*}} - \mathrm{negl}(\kappa)$. Thus, if the forger $\mathcal{F}$ can break the EUF-CMA security of $\mathcal{SIG}_4$ with probability ϵ in the real game, then it will also output a valid forgery $(M^*, \mathbf{e}^*)$ in the simulated game with probability at least $(\epsilon - Q^2(\frac{Q}{2^{j^*}})^v) \cdot \delta \cdot (\frac{1}{2^{j^*}} - \mathrm{negl}(\kappa)) = \frac{\epsilon}{2^{j^*}\cdot 16nv^2} - \mathrm{negl}(\kappa) \ge \frac{\epsilon}{64\cdot Q\cdot nv^2} - \mathrm{negl}(\kappa)$.

Now, it suffices to show that $\hat{\mathbf{e}} = (\mathbf{I}_{\bar{m}} \| \hat{\mathbf{R}}^*)\mathbf{e}^*$ is a valid solution to the $\mathrm{ISIS}_{q,\bar{m},\bar{\beta}}$ instance $(\mathbf{A}, \mathbf{u})$. By the verification algorithm, we have that $\mathbf{A}_{M^*,\mathbf{t}^*}\mathbf{e}^* = \mathbf{u}$ and $\|\mathbf{e}^*\| \leq s\sqrt{m}$. Since $s_1(\mathbf{R}_i) \leq \sqrt{m}\cdot\omega(\sqrt{\log n})$ by Lemma 3 and $s_1(\mathbf{R}_{M^*}) \leq \beta = \tilde{O}(n^{2.5})$ by Theorem 3, we have that $s_1(\hat{\mathbf{R}}^*) \leq \tilde{O}(n^{2.5})$ and that $\|\hat{\mathbf{e}}\| \leq \tilde{O}(n^{2.5}) \cdot s\sqrt{m} = \tilde{O}(n^{5.5}) = \bar{\beta}$. This completes the proof. □

Remark 3. Our construction of $\mathcal{SIG}_4$ essentially supports to use a tag of length up to $\ell = n - 1$. In particular, we still have $O(\log n)$ number of matrices in the verification key and $\tilde{O}(n)$ bits in the signature for the choice of $\ell = n-1$, but the ISIS parameter $\bar{\beta}$ for the security proof of Theorem 10 will increase by roughly a factor of n. We set $\ell = \omega(\log n)$ because 1) it is sufficient to handle any adversary making a polynomially bounded number $Q = \mathsf{poly}(n)$ of signing queries, and 2) it allows to obtain a better asymptotic parameter for the underlying ISIS problem. We also note that it is possible to make a trade-off between the ISIS parameter and the reduction loss by using a different way to parse the tag space $\{0,1\}^\ell$ as for the confined guessing technique in [9].

5.2 A Fully-secure Short Signature on Ideal Lattices

Following the same idea of $\mathcal{SIG}_4$, we can also construct a fully secure signature scheme with constant verification keys from the Type-III PHF. Formally, let $n, k \in \mathbb{Z}$ be some polynomials in the security parameter κ, and let $q = 3^k, w = \lceil \log_2 q \rceil + 2, \bar{m} = w + k, m = \bar{m} + k$ and $s = \tilde{O}(n^{2.5}) \in \mathbb{R}$. Let $R_q = \mathbb{Z}_q[x]/(x^n+1)$, and let $\ell = \omega(\log n) < \frac{n}{2}$. For any $t \in \{0,1\}^\ell$, let $t_j \in \{0,1\}^j$ be the binary string consisting of the first j bits of t (which means that $t_\ell = t$). Let $d \geq 2$ be a factor of $2n$ such that $d/2 \leq \ell < d$. For any $t \in \{0,1\}^{\ell'}$ with $\ell' < n$, we naturally treat it as a polynomial of degee at most $\ell' - 1$ in R_q with the canonical coefficient embedding. Let $\mathcal{H} = (\mathcal{H}.\mathrm{Gen}_R, \mathcal{H}.\mathrm{Eval}_R)$ be the improved Type-III PHF given in Definition 8. Our signature scheme $\mathcal{SIG}_5 = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ is defined as follows.

$\mathsf{KeyGen}(1^\kappa)$**:** Given a security parameter κ as input, the key generation algorithm first computes $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}_R(1^n, 1^{\bar{m}}, q, 1)$ such that $\mathbf{A} \in R_q^{1\times\bar{m}}, \mathbf{R} = R_q^{w\times k}$. Randomly choose $\mathbf{A}_0, \mathbf{A}_1 \leftarrow R_q^{1\times k}$ and $u \leftarrow R_q$. Finally, compute $K \leftarrow \mathcal{H}.\mathrm{Gen}_R(1^\kappa)$, and return $vk = (\mathbf{A}, \mathbf{A}_0, \mathbf{A}_1, u, K)$ and $sk = (vk, \mathbf{R})$.

$\mathsf{Sign}(sk, M \in \{0,1\}^n)$**:** Given the secret key sk and a message M as inputs, randomly choose a bit sting $t \in \{0,1\}^\ell$, let $t_j \in \{0,1\}^j$ be the binary string consisting of the first j bits of t. Then, compute $\hat{\mathbf{A}}_j = \mathsf{Rcomp}_{\ell,d}(\mathbf{A}_0, d, j)$ for all $1 \leq j \leq \ell$. Next, compute

$$\mathbf{A}_{M,t} = (\mathbf{A} \| (\mathbf{A}_1 + \sum_{j=1}^{\ell} \hat{\mathbf{A}}_j \cdot \mathbf{G}_3^{-1}(xt_j\mathbf{G}_3) + \mathrm{H}_K(M))) \in R_q^{1\times m},$$

where $\mathrm{H}_K(M) = \mathcal{H}.\mathrm{Eval}_R(K, M) \in R_q^{1\times k}$. Finally, compute and return the signature $\sigma = (\mathbf{e}, t)$, where $\mathbf{e} \leftarrow \mathsf{SampleD}_R(\mathbf{R}, \mathbf{A}_{M,t}, 1, u, s) \in R_q^m$.

$\mathsf{Verify}(vk, M, \sigma)$**:** Given vk, message M and $\sigma = (\mathbf{e}, t)$ as inputs, compute $\hat{\mathbf{A}}_j = \mathsf{Rcomp}_{\ell,d}(\mathbf{A}_0, d, j)$ for all $1 \leq j \leq \ell$. Next, compute

$$\mathbf{A}_{M,t} = (\mathbf{A} \| (\mathbf{A}_1 + \sum_{j=1}^{\ell} \hat{\mathbf{A}}_j \cdot \mathbf{G}_3^{-1}(xt_j\mathbf{G}_3) + \mathrm{H}_K(M))) \in R_q^{1\times m},$$

where $\mathrm{H}_K(M) = \mathcal{H}.\mathrm{Eval}_R(K, M) \in R_q^{1\times k}$. Return 1 if $\|\mathbf{e}\| \leq s\sqrt{mn}$ and $\mathbf{A}_{M,t}\mathbf{e} = u$. Otherwise, return 0.

Since $\mathbf{R}$ is a $\mathbf{G}$-trapdoor of $\mathbf{A}$, by padding with zero rows it can be extended to a $\mathbf{G}$-trapdoor for $\mathbf{A}_{M,t}$ with the same quality $s_1(\mathbf{R}) \leq (\sqrt{nw} + \sqrt{nk} + \omega(\sqrt{n \log n})) \cdot \omega(\sqrt{\log nw})$. Since $s = \tilde{O}(n^{2.5}) > s_1(\mathbf{R}) \cdot \omega(\sqrt{\log n})$, the vector $\mathbf{e}$ output by $\mathsf{SampleD}_R$ follows the distribution $(D_{R,s})^m$ satisfying $\mathbf{A}_{M,t}\mathbf{e} = u$. In other words, $\|\mathbf{e}\| \leq s\sqrt{mn}$ holds with overwhelming probability by Lemma 2. This shows that $\mathcal{SIG}_5$ is correct.

Note that if we set $v = \omega(\log n)$ in Definition 8, the key K only has 3 vectors of $R_q^{1\times k}$ and each signature consists of a vector in R_q^m plus a short ℓ-bit tag. Set $w = \lceil \log_2 q \rceil + 2 \leq 2k$, we have the verification key having at most 8 vectors in $R_q^{1\times k}$. We have the following theorem for security.

Theorem 11. *Let $\ell, \bar{m}, n, q, v \in \mathbb{Z}$ be polynomials in the security parameter κ. Let $\ell = \omega(\log n), v = \omega(\log n)$. Then, if there exists a PPT forger $\mathcal{F}$ breaking the* EUF-CMA *security of $\mathcal{SIG}_5$ with non-negligible probability ϵ and making at most $Q = \mathsf{poly}(n)$ signing queries, there exists an algorithm $\mathcal{B}$ solving the* $\mathrm{ISIS}_{q,\bar{m},\bar{\beta}}$ *problem over R for $\bar{\beta} = \tilde{O}(n^{5.5})$ with probability at least $\epsilon' \geq \frac{\epsilon}{64\cdot Q\cdot nv^2} - \mathrm{negl}(\kappa) = \frac{\epsilon}{Q\cdot\tilde{O}(n)}$.*

Proof. The proof is very similar to that of Theorem 9. Formally, we now construct an algorithm $\mathcal{B}$, which simulates the attack environment for $\mathcal{F}$, and solves the $\mathrm{ISIS}_{q,\bar{m},\bar{\beta}}$ problem over R with probability at least $\frac{\epsilon}{Q\cdot\tilde{O}(n)}$. Let $1 \leq j^* \leq \ell$ be the smallest integer such that $Q/2^{j^*} \leq 1/2 < Q/2^{j^*-1}$. $\mathcal{B}$ first randomly chooses a bit sting $t' \in \{0,1\}^{j^*}$, and hopes that $\mathcal{F}$ will output a forged signature with tag t^* such that $t^*_{j^*} = t'$. Then, $\mathcal{B}$ simulates the EUF-CMA game as follows:

KeyGen. Given an $\mathrm{ISIS}_{q,\bar{m},\bar{\beta}}$ challenge instance $(\mathbf{A}, u) \in R_q^{1\times\bar{m}} \times R_q$, the algorithm $\mathcal{B}$ first randomly chooses $\mathbf{R}_0, \mathbf{R}_1 \leftarrow (D_{R,\omega(\sqrt{\log n\bar{m}})})^{\bar{m}\times k}$, and computes $\mathbf{A}_0 = \mathbf{A}\mathbf{R}_0 + x^{\frac{2n}{d}j^*}\mathbf{G}_3$, $\mathbf{A}_1 = \mathbf{A}\mathbf{R}_1 - xt'\mathbf{G}_3$. Then, compute $(K', td) \leftarrow \mathcal{H}.\mathrm{TrapGen}_R(1^\kappa, \mathbf{A}, \mathbf{G}_3)$ as in Theorem 5. Finally, set $vk = (\mathbf{A}, \mathbf{A}_0, \mathbf{A}_1, u, K')$ and keep $(\mathbf{R}_0, \mathbf{R}_1, td)$ private.

Signing. Given a message M, the algorithm $\mathcal{B}$ first randomly choose a bit sting $t \in \{0,1\}^\ell$, let $t_j \in \{0,1\}^j$ be the binary string consisting of the first j bits of t. If t_{j^*} has been used in answering the signatures for more than v messages, $\mathcal{B}$ aborts. Otherwise, compute $\hat{\mathbf{A}}_j = \mathsf{Rcomp}_{\ell,d}(\mathbf{A}_0, d, j)$ for all $1 \leq j \leq \ell$.

Next, compute

$$\mathbf{A}_{M,t} = (\mathbf{A}\|(\mathbf{A}_1 + \sum_{j=1}^{\ell} \hat{\mathbf{A}}_j \cdot \mathbf{G}_3^{-1}(xt_j\mathbf{G}_3) + \mathrm{H}_{K'}(M))) \in R_q^{1\times m},$$

where $\mathrm{H}_K(M) = \mathcal{H}.\mathrm{Eval}_R(K, M) \in R_q^{1\times k}$. For all $1 \leq j \leq \ell$, let $(\hat{\mathbf{R}}_j, \hat{h}_j) = \mathsf{TRcomp}_{\ell,d}(\mathbf{A}_0, \mathbf{R}_0, d, j)$ and $(\mathbf{R}_M, h_M) = \mathcal{H}.\mathrm{TrapEval}_R(td, K', M)$. By the proof of Theorem 4 we have $\hat{h}_j = 1$ if and only if $j = j^*$, and 0 otherwise. Thus, we have

$$\mathbf{A}_{M,t} = (\mathbf{A}\|\mathbf{A}(\mathbf{R}_1 + \sum_{j=1}^{\ell} \hat{\mathbf{R}}_j \mathbf{G}_3^{-1}(xt_j\mathbf{G}_3) + \mathbf{R}_M) + (x(t_{j^*} - t') + h_M)\mathbf{G}_3).$$

$\mathcal{B}$ distinguishes the following two cases:

- $t_{j^*} \neq t'$ or $(t_{j^*} = t' \wedge h_M \neq 0)$: In both cases, we have that $\hat{h} = x(t_{j^*} - t') + h_M$ is invertible by Lemma 6, because $\hat{h}$ has coefficients in $\{-1, 0, 1\}$ (note that $h_M \in \{0, 1\}$) and has degree at most $\ell < n/2$. In other words, $\hat{\mathbf{R}} = \mathbf{R}_0 + \sum_{j=1}^{\ell} \hat{\mathbf{R}}_j \mathbf{G}_3^{-1}(xt_j\mathbf{G}_3) + \mathbf{R}_M$ is a $\mathbf{G}$-trapdoor for $\mathbf{A}_{M,t}$. Since $s_1(\mathbf{R}_0) \leq (\sqrt{w} + \sqrt{k} + \omega(\sqrt{\log n})) \cdot \omega(\sqrt{\log n\bar{m}})$ by Lemma 7, $s_1(\hat{\mathbf{R}}_j) \leq \tilde{O}(n^{1.5})$ by the proof of Theorem 4, and $s_1(\mathbf{R}_M) \leq \tilde{O}(n^{2.5})$, we have $s_1(\hat{\mathbf{R}}) \leq \tilde{O}(n^{2.5})$. Then, compute $\mathbf{e} \leftarrow \mathsf{SampleD}_R(\hat{\mathbf{R}}, \mathbf{A}_{M,t}, \hat{h}, u, s)$, and return the signature $\sigma = (\mathbf{e}, t)$. If we set an appropriate $s = \tilde{O}(n^{2.5}) \geq s_1(\hat{\mathbf{R}}) \cdot \omega(\sqrt{\log n})$, then $\mathcal{B}$ can generate a valid signature on M with overwhelming probability by Proposition 2.
- $t_{j^*} = t' \wedge h_M = 0$: $\mathcal{B}$ aborts.

Forge. After making at most Q signing queries, $\mathcal{F}$ outputs a forged signature $\sigma^* = (\mathbf{e}^*, t^*)$ on message $M^* \in \{0,1\}^n$ such that $\|\mathbf{e}^*\| \leq s\sqrt{mn}$ and $\mathbf{A}_{M^*,t^*}\mathbf{e}^* = u$, where

$$\mathbf{A}_{M^*,t^*} = (\mathbf{A}\|(\mathbf{A}_1 + \sum_{j=1}^{\ell} \hat{\mathbf{A}}_j \cdot \mathbf{G}_3^{-1}(xt_j^*\mathbf{G}_3) + \mathrm{H}_{K'}(M^*))) \in R_q^{1\times m}.$$

For all $1 \leq j \leq \ell$, let $(\hat{\mathbf{R}}_j, \hat{h}_j) = \mathsf{TRcomp}_{\ell,d}(\mathbf{A}_0, \mathbf{R}_0, d, j)$ and $(\mathbf{R}_{M^*}, h_{M^*}) = \mathcal{H}.\mathrm{TrapEval}_R(td, K', M^*)$. The algorithm $\mathcal{B}$ aborts the simulation if $t_{j^*}^* \neq t'$ or $h_{M^*} \neq 0$. Else, we have

$$\mathbf{A}_{M^*,t^*} = (\mathbf{A}\|\mathbf{A}(\mathbf{R}_1 + \sum_{j=1}^{\ell} \hat{\mathbf{R}}_j \cdot \mathbf{G}_3^{-1}(xt_j^*\mathbf{G}_3) + \mathbf{R}_{M^*})) = (\mathbf{A}\|\mathbf{A}\hat{\mathbf{R}}),$$

where $\hat{\mathbf{R}} = \mathbf{R}_0 + \sum_{j=1}^{\ell} \hat{\mathbf{R}}_j \cdot \mathbf{G}_3^{-1}(xt_j^*\mathbf{G}_3) + \mathbf{R}_{M^*}$. Finally, $\mathcal{B}$ outputs $\hat{\mathbf{e}} = (\mathbf{I}_{\bar{m}}\|\hat{\mathbf{R}})\mathbf{e}^*$ as its own solution.

By the definition of the $\mathrm{ISIS}_{q,\bar{m},\bar{\beta}}$ problem over R, $(\mathbf{A}, u)$ is uniformly distributed over $R_q^{1\times\bar{m}} \times R_q$. Since $\mathbf{R}_0 \leftarrow (D_{R,\omega(\sqrt{\log n\bar{m}})})^{\bar{m}\times k}$, we have that $\mathbf{A}_0 \in$

$R_q^{1\times k}$ is statistically close to uniform over $R_q^{1\times k}$ by Lemma 9. In addition, by Theorem 5 the simulated key K' is statistically close to the real key K. Thus, the distribution of the simulated verification key vk is statistically close to that of the real one.

Let $M_1, \dots, M_u$ be all the messages in the signing queries that $\mathcal{B}$ happens to use $t' \in \{0,1\}^{j^*}$ in generating the signatures, and let $(\mathbf{R}_{M_i}, h_{M_i}) = \mathcal{H}.\mathrm{TrapEval}(td, K', M_i)$ for $i \in \{1,\dots,u\}$. Then, $\mathcal{B}$ will abort in the simulation if and only if either of the following two conditions hold:

- Some $t_{j^*} \in \{0,1\}^{j^*}$ has been used in generating the signatures for more than v messages;
- $h_{M_i} = 0$ is not invertible for some $i \in \{1,\dots,u\}$, or $h_{M^*} \neq \mathbf{0}$, or $t^*_{j^*} \neq t'$.

Note that $\mathcal{B}$ always randomly chooses a tag $t \leftarrow \{0,1\}^\ell$ for each signing message, the probability that the same $t_{j^*} \in \{0,1\}^{j^*}$ is used in generating the signatures for more than v messages is less than $Q^2 \cdot (\frac{Q}{2^{j^*}})^v$ by a similar analysis in [35], which is negligible by our choices of $\frac{Q}{2^{j^*}} \leq 1/2$ and $v = \omega(\log n)$. In particular, the probability that $\mathcal{B}$ will use $t' \in \{0,1\}^{j^*}$ in generating the signatures for $u \geq v$ messages is also negligible. Conditioned on $u \leq v$, the probability that $h_{M_i} = 1$ for all $i \in \{1,\dots,u\}$ and $h_{M^*} = 0$ (using the fact that $M^* \notin \{M_1,\dots,M_u\}$) is at least $\delta = \frac{1}{16nv^2} - \mathrm{negl}(\kappa)$ by Theorem 5. Note that t' is randomly chosen and is statistically hidden from $\mathcal{F}$, we have that the probability $\Pr[t^*_{j^*} = t']$ is at least $\frac{1}{2^{j^*}} - \mathrm{negl}(\kappa)$. Thus, if the forger $\mathcal{F}$ can break the EUF-CMA security of $\mathcal{SIG}_5$ with probability ϵ in the real game, then it will also output a valid forgery $(M^*, \mathbf{e}^*)$ in the simulated game with probability at least $(\epsilon - Q^2(\frac{Q}{2^{j^*}})^v) \cdot \delta \cdot (\frac{1}{2^{j^*}} - \mathrm{negl}(\kappa)) = \frac{\epsilon}{2^{j^*} \cdot 16nv^2} - \mathrm{negl}(\kappa) \geq \frac{\epsilon}{64 \cdot Q \cdot nv^2} - \mathrm{negl}(\kappa)$.

Now, it suffices to show that $\hat{\mathbf{e}} = (\mathbf{I}_{\bar{m}} \| \hat{\mathbf{R}}^*)\mathbf{e}^*$ is a valid solution to the $\mathrm{ISIS}_{q,\bar{m},\bar{\beta}}$ instance $(\mathbf{A}, \mathbf{u})$. By the conditions in the verification algorithm, we have that $\mathbf{A}_{M^*,\mathbf{t}^*}\mathbf{e}^* = \mathbf{u}$ and $\|\mathbf{e}^*\| \leq s\sqrt{m}$. Since $s_1(\mathbf{R}_1) \leq \sqrt{m} \cdot \omega(\sqrt{\log n})$ by Lemma 3, $s_1(\hat{\mathbf{R}}_j) \leq \tilde{O}(n^{1.5})$ by the proof of Theorem 4 and $s_1(\mathbf{R}_{M^*}) \leq \beta = \tilde{O}(n^{2.5})$ by Theorem 5, we have that $s_1(\hat{\mathbf{R}}^*) \leq \tilde{O}(n^{2.5})$ and that $\|\hat{\mathbf{e}}\| \leq \tilde{O}(n^{2.5}) \cdot s\sqrt{m} = \tilde{O}(n^{5.5}) = \bar{\beta}$. This completes the proof. □

Remark 4. Similar to $\mathcal{SIG}_4$, our construction of $\mathcal{SIG}_5$ also supports to use a tag of length up to $\ell = n/2 - 1$ with a price of increasing the underlying ISIS parameter $\bar{\beta}$ by a factor of at most n in the security proof.

6 IBE from Lattice-based PHFs

In this section, we give a generic IBE scheme from Lattice-based PHFs.

6.1 A Generic IBE scheme from Lattice-based PHFs

Let integers $n, \bar{n}, m', \ell, v, \beta, q$ be polynomials in the security parameter κ, and let $k = \lceil \log_2 q \rceil$. Let ring $\mathcal{R}$ be either the integer ring $\mathbb{Z}$ or the polynomial ring

$\mathbb{Z}[x]/(x^n+1)$. Let $\mathcal{R}_q = \mathcal{R}/(q\mathcal{R})$ be the quotient ring. Let $n = 1$ if $\mathcal{R} = \mathbb{Z}$. Let $\mathcal{H} = (\mathcal{H}.\mathrm{Gen}_{\mathcal{R}}, \mathcal{H}.\mathrm{Eval}_{\mathcal{R}})$ be a PHF defined over ring $\mathcal{R}$ from $\{0,1\}^\ell$ to $\mathbb{Z}_q^{\bar{n}\times m'}$. Let $n = 1$ if $\mathcal{R} = \mathbb{Z}$. We set the user identity space as $\{0,1\}^\ell$ and the message space as $\mathcal{R}_2^{\bar{n}}$. Let integers $\bar{m} = O(\bar{n}\log q), m = \bar{m} + m'$, $\alpha \in \mathbb{R}$, and large enough $s > \max(\beta, \sqrt{m})\cdot\omega(\sqrt{\log n\bar{n}})$ be the system parameters. Our generic IBE scheme $\mathcal{IBE} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ is defined as follows.

$\mathsf{Setup}(1^\kappa)$**:** Given a security parameter κ as input, the algorithm first computes $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}_{\mathcal{R}}(1^{\bar{n}}, 1^{\bar{m}}, q, \mathbf{I}_{\bar{n}})$ such that $\mathbf{A} \in \mathcal{R}_q^{\bar{n}\times\bar{m}}$, $\mathbf{R} = \mathcal{R}_q^{(\bar{m}-\bar{n}k)\times\bar{n}k}$. Randomly choose $\mathbf{U} \leftarrow \mathcal{R}_q^{\bar{n}\times\bar{n}}$, and compute $K \leftarrow \mathcal{H}.\mathrm{Gen}_{\mathcal{R}}(1^\kappa)$. Finally, return $(mpk, msk) = ((\mathbf{A}, K, \mathbf{U}), \mathbf{R})$.

$\mathsf{Extract}(msk, id \in \{0,1\}^\ell)$**:** Given msk and a user identity id as inputs, compute $\mathbf{A}_{id} = (\mathbf{A}\|\mathrm{H}_K(id)) \in \mathcal{R}_q^{\bar{n}\times m}$, where $\mathrm{H}_K(id) = \mathcal{H}.\mathrm{Eval}_{\mathcal{R}}(K, id) \in \mathcal{R}_q^{\bar{n}\times m'}$. Then, compute $\mathbf{E}_{id} \leftarrow \mathsf{SampleD}_{\mathcal{R}}(\mathbf{R}, \mathbf{A}_{id}, \mathbf{I}_n, \mathbf{U}, s)$, and return $sk_{id} = \mathbf{E}_{id} \in \mathcal{R}^{m\times\bar{n}}$.

$\mathsf{Enc}(mpk, id \in \{0,1\}^\ell, M \in \mathcal{R}_2^{\bar{n}})$**:** Given mpk, id and plaintext M as inputs, compute $\mathbf{A}_{id} = (\mathbf{A}\|\mathrm{H}_K(id)) \in \mathcal{R}_q^{\bar{n}\times m}$, where $\mathrm{H}_K(id) = \mathcal{H}.\mathrm{Eval}_{\mathcal{R}}(K, id) \in \mathcal{R}_q^{\bar{n}\times m'}$. Then, randomly choose $\mathbf{s} \leftarrow \mathcal{R}_q^{\bar{n}}$, $\mathbf{x}_0 \leftarrow (D_{\mathcal{R},\alpha q})^{\bar{n}}, \mathbf{x}_1 \leftarrow (D_{\mathcal{R},\alpha q})^{\bar{m}}$, and $\mathbf{x}_2 \leftarrow (D_{\mathcal{R},2\alpha\beta q})^{m'}$. Finally, compute and return the ciphertext $\mathbf{C} = (\mathbf{c}_0, \mathbf{c}_1)$, where

$$\mathbf{c}_0 = \mathbf{U}^T\mathbf{s} + \mathbf{x}_0 + \frac{q}{2}M, \qquad \mathbf{c}_1 = \mathbf{A}_{id}^T\mathbf{s} + \begin{pmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{pmatrix}.$$

$\mathsf{Dec}(sk_{id}, \mathbf{C})$**:** Given $sk_{id} = \mathbf{E}_{id}$ and a ciphertext $\mathbf{C} = (\mathbf{c}_0, \mathbf{c}_1)$ under identity id as inputs, compute $\mathbf{b} = \mathbf{c}_0 - \mathbf{E}_{id}^T\mathbf{c}_1 \in \mathcal{R}_q^{\bar{n}}$. Then, compute $M = \lfloor\frac{2}{q}\mathbf{b}\rceil \bmod 2 \in \mathcal{R}_2^{\bar{n}}$. Finally, return the plaintext $M \in \mathcal{R}_2^{\bar{n}}$.

By the correctness of $\mathsf{SampleD}_{\mathcal{R}}$ we know that $\mathbf{A}_{id}\mathbf{E}_{id} = \mathbf{U}$ and $\|\mathbf{E}_{id}\| \leq s\sqrt{m\bar{n}}$ hold with overwhelming probability. One can easily show that the decryption algorithm is correct for appropriate choices of parameters.

6.2 The Security Proof

For security, we show that under the LWE assumption over $\mathcal{R}$, our generic IBE scheme $\mathcal{IBE}$ is provably secure in the standard model.

Theorem 12. *Let $\bar{n}, q, m' \in \mathbb{Z}$ and $\alpha, \beta \in \mathbb{R}$ be polynomials in the security parameter κ. Let $\mathcal{H} = (\mathcal{H}.\mathrm{Gen}_{\mathcal{R}}, \mathcal{H}.\mathrm{Eval}_{\mathcal{R}})$ be any $(1, \mathsf{poly}, \beta, \gamma, \delta)$-*PHF *defined over $\mathcal{R}$ from $\{0,1\}^\ell$ to $\mathcal{R}_q^{\bar{n}\times m'}$, where $\gamma = \mathrm{negl}(\kappa)$ and $\delta > 0$ is noticeable. Then, if there exists a PPT adversary $\mathcal{A}$ breaking the* IND-ID-CPA security of $\mathcal{IBE}$ with non-negligible advantage ϵ and making at most polynomially bounded number Q of user private key queries, there exists an algorithm $\mathcal{B}$ solving the $\mathrm{LWE}_{q,\alpha}$ problem over $\mathcal{R}$ with advantage at least $\epsilon' \geq \epsilon\delta/2 - \mathrm{negl}(\kappa)$.

Moreover, if $\mathcal{H} = (\mathcal{H}.\mathrm{Gen}_{\mathcal{R}}, \mathcal{H}.\mathrm{Eval}_{\mathcal{R}})$ is a $(1, v, \beta, \gamma, \delta)$-PHF for some prior fixed polynomial $v = \mathsf{poly}(n)$, then the resulting IBE scheme is secure against any PPT adversary making at most $Q \leq v$ user private key queries (namely, it satisfies the IND-qID-CPA security).

Proof. In the following, we use a sequence of games from Games 0 to 6: Game 0 is exactly the real security game as in Definition 3 where the challenger honestly encrypts the challenge plaintext, while Game 6 is a random game where the challenge ciphertext is independent from the challenge plaintext. The security is established by showing that if $\mathcal{A}$ can succeed in Game 0 with non-negligible advantage ϵ, then it can also succeed in Game 6 with non-negligible advantage, which is contradictory to the fact that Game 6 is a random game. Let $\mathbf{B} \in \mathbb{Z}_q^{\bar{n} \times m'}$ be any trapdoor matrix that allows to efficiently sample short vector $\mathbf{v}$ satisfying $\mathbf{B}\mathbf{v} = \mathbf{u}$ for any $\mathbf{u} \in \mathcal{R}_q^{\bar{n}}$, by using the trapdoor delegation techniques in [2].

Game 0. The challenger $\mathcal{C}$ honestly simulates the IND-ID-CPA security game for $\mathcal{A}$ as follows:

Setup. First compute $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}_{\mathcal{R}}(1^{\bar{n}}, 1^{\bar{m}}, q, \mathbf{I}_{\bar{n}})$ such that $\mathbf{A} \in \mathcal{R}_q^{\bar{n} \times \bar{m}}$, $\mathbf{R} = \mathcal{R}_q^{(\bar{m} - \bar{n}k) \times \bar{n}k}$. Then, randomly choose $\mathbf{U} \leftarrow \mathcal{R}_q^{\bar{n} \times \bar{n}}$, and compute $K \leftarrow \mathcal{H}.\mathrm{Gen}_{\mathcal{R}}(1^{\kappa})$. Finally, send the master public key $mpk = (\mathbf{A}, K, \mathbf{U})$ to the adversary $\mathcal{A}$, and keep the master secret key $\mathbf{R}$ private.

Phase 1. Upon receiving the user private key query with identity $id \in \{0,1\}^{\ell}$, compute the hash value $\mathbf{A}_{id} = (\mathbf{A} \| \mathrm{H}_K(id)) \in \mathcal{R}_q^{\bar{n} \times m}$, where $\mathrm{H}_K(id) = \mathcal{H}.\mathrm{Eval}(K, id) \in \mathcal{R}_q^{\bar{n} \times m'}$. Then, compute $\mathbf{E}_{id} \leftarrow \mathsf{SampleD}_{\mathcal{R}}(\mathbf{R}, \mathbf{A}_{id}, \mathbf{I}_{\bar{n}}, \mathbf{U}, s)$, and send the user private key $sk_{id} = \mathbf{E}_{id} \in \mathcal{R}^{m \times \bar{n}}$ to the adversary $\mathcal{A}$.

Challenge. At some time, the adversary $\mathcal{A}$ outputs a challenge identity id^* and a pair of challenge plaintexts $(M_0, M_1) \in \mathcal{R}_2^{\bar{n}} \times \mathcal{R}_2^{\bar{n}}$ with the restriction that it never obtains the user private key of id^* in Phase 1. The challenger $\mathcal{C}$ randomly chooses a bit $b^* \leftarrow \{0,1\}$, $\mathbf{s} \leftarrow \mathcal{R}_q^{\bar{n}}$, $\mathbf{x}_0 \leftarrow (D_{\mathcal{R},\alpha q})^{\bar{n}}, \mathbf{x}_1 \leftarrow (D_{\mathcal{R},\alpha q})^{\bar{m}}$, and $\mathbf{x}_2 \leftarrow (D_{\mathcal{R},2\alpha\beta q})^{m'}$. Then, it sets $\mathbf{C}_{b^*} = (\mathbf{c}_0^*, \mathbf{c}_1^*)$, where

$$\mathbf{c}_0^* = \mathbf{U}^T \mathbf{s} + \mathbf{x}_0 + \frac{q}{2} M_{b^*}, \quad \mathbf{c}_1^* = \mathbf{A}_{id^*}^T \mathbf{s} + \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix},$$

where $\mathbf{A}_{id^*} = (\mathbf{A} \| \mathrm{H}_K(id^*)) \in \mathcal{R}_q^{\bar{n} \times m}$ and $\mathrm{H}_K(id^*) = \mathcal{H}.\mathrm{Eval}(K, id^*) \in \mathcal{R}_q^{\bar{n} \times m'}$. Finally, it sends the challenge ciphertext $\mathbf{C}_{b^*}$ to the adversary $\mathcal{A}$.

Phase 2. $\mathcal{A}$ can adaptively make more user private key queries with any identity $id \neq id^*$. The challenger $\mathcal{C}$ responds as in Phase 1.

Guess. Finally, $\mathcal{A}$ outputs a guess $b \in \{0,1\}$. If $b = b^*$, the challenger $\mathcal{C}$ outputs 1, else outputs 0.

Denote F_i as the event that $\mathcal{C}$ outputs 1 in Game i for $i \in \{0, 1, \ldots, 5\}$.

Lemma 10. $|\Pr[F_0] - \frac{1}{2}| = \epsilon$.

Proof. This lemma immediately follows from the fact that $\mathcal{C}$ honestly simulates the attack environment for $\mathcal{A}$, and outputs 1 if and only if $b = b^*$. □

Game 1. This game is identical to Game 0 except that $\mathcal{C}$ changes the guess phase as follows.

Guess. Finally, the adversary $\mathcal{A}$ outputs a guess $b \in \{0,1\}$. Let $id_1, \ldots, id_Q$ be all the identities in the user private queries, and let id^* be the challenge identity. Denote $I^* = \{id_1, \ldots, id_Q, id^*\}$, the challenger $\mathcal{C}$ computes $(td, K') \leftarrow \mathcal{H}.\mathrm{TrapGen}_{\mathcal{R}}(1^\kappa, \mathbf{A}, \mathbf{B})$ and $(\mathbf{R}_{id_i}, \mathbf{S}_{id_i}) = \mathcal{H}.\mathrm{TrapEval}_{\mathcal{R}}(td, K', id_i)$. Then, it defines the following function

$$\tau(td, K', I^*) = \begin{cases} 0, & \text{if } \mathbf{S}_{id^*} = \mathbf{0}, \text{ and } \mathbf{S}_{id_i} \text{ is invertible for all } i \in \{1, \ldots, Q\} \\ 1, & \text{otherwise,} \end{cases}$$

Then, $\mathcal{C}$ proceeds the following steps:

1. **Abort check**: If $\tau(td, K', I^*) = 1$, the challenger $\mathcal{C}$ aborts the game, and outputs a uniformly random bit.
2. **Artificial abort**: Fixing $I^* = \{id_1, \ldots, id_Q, id^*\}$, let p be the probability $p = \Pr[\tau(td', K', I^*) = 0]$ over the random choice of (td', K'). Then, the challenger $\mathcal{C}$ samples $O(\epsilon^2 \log(\epsilon^{-1})\delta^{-1}\log(\delta^{-1}))$ times the probability p by independently running $(td', K') \leftarrow \mathcal{H}.\mathrm{TrapGen}_{\mathcal{R}}(1^\kappa, \mathbf{A}, \mathbf{B})$ and evaluating $\tau(td', K, I^*)$ to compute an estimate p'.[10] Let δ be the parameter for the well-distributed hidden matrices property of $\mathcal{H}$, if $p' > \delta$, the challenger $\mathcal{C}$ aborts with probability $\frac{p'-\delta}{p'}$, and outputs a uniformly random bit.

Finally, if $b = b^*$, the challenger $\mathcal{C}$ outputs 1, else outputs 0.

Remark 5. As in [54,8,2,16,28], this seemingly meaningless artificial abort stage is necessary for our later refinements. Looking ahead, in the following games the challenger $\mathcal{C}$ can continue the simulation only when the identities $id_1, \ldots, id_Q, id^*$ will not cause an abort (in the abort check stage). Since the success probability of the adversary $\mathcal{A}$ might be correlated with the probability that $\mathcal{C}$ aborts, it becomes complicate when we try to rely the success probability of $\mathcal{C}$ (in solving the underlying LWE problems) on the success probability of the adversary $\mathcal{A}$ (in attacking the IBE scheme). In [54], Waters introduced the artificial abort to force the probability that $\mathcal{C}$ aborts to be independent of $\mathcal{A}$'s particular queries. In certain cases, Bellare and Ristenpart [8] showed that the artificial abort can be avoided. Because our construction uses general lattice-based PHFs as a "blackbox", we opt for the Waters approach and introduce an artificial abort. Besides, we clarify that there is no artificial abort involved in our generic signature scheme because any valid forgery can be publicly checked by the challenger $\mathcal{C}$. Similar argument can be found in [54].

Lemma 11. *If $\mathcal{H}$ is a $(1, \mathsf{poly}, \beta, \gamma, \delta)$-PHF, then $|\Pr[F_1] - \frac{1}{2}| \geq \frac{1}{2}\epsilon\delta$.*

[10] In general, the sampling procedure generally makes the running time of $\mathcal{C}$ dependent on the success advantage ϵ of $\mathcal{A}$, but for concrete PHFs (e.g., the construction in Theorem 3), it is possible to directly calculate the probability p.

Proof. Let $\mathcal{QID} = (\{0,1\}^\ell)^{Q+1}$ be the set of all $Q+1$ tuples of identities. Let $\mathcal{Q}(I)$ be the event that the adversary $\mathcal{A}$ uses the first Q identities in $I = \{id_1, \ldots, id_Q, id^*\} \in \mathcal{QID}$ for user private key queries, and the last one for the challenge identity. Let $F_i(I) \subseteq \mathcal{Q}(I)$ be the event that the challenger $\mathcal{C}$ outputs 1 in Game i when $\mathcal{Q}(I)$ happens, where $i \in \{0,1\}$. Let $\mathcal{E}$ be the event that $\mathcal{C}$ aborts in Game 1. Then, by the definition we have the following facts:

$$\begin{aligned}
&\textstyle\sum_{I\in\mathcal{QID}} \Pr[\mathcal{Q}(I)] = 1\\
&\Pr[F_i] = \textstyle\sum_{I\in\mathcal{QID}} \Pr[F_i(I)]\\
&\Pr[F_i] = \Pr[F_i \wedge \mathcal{E}] + \Pr[F_i \wedge \neg\mathcal{E}]\\
&\Pr[\mathcal{Q}(I)] = \Pr[\mathcal{Q}(I) \wedge \mathcal{E}] + \Pr[\mathcal{Q}(I) \wedge \neg\mathcal{E}]
\end{aligned}$$

Besides, by the description of Game 1, we have that $\Pr[F_1(I) \wedge \mathcal{E}] = \frac{1}{2}\Pr[\mathcal{Q}(I) \wedge \mathcal{E}]$ and $\Pr[F_1(I) \wedge \neg\mathcal{E}] = \Pr[F_0(I) \wedge \neg\mathcal{E}] = \Pr[F_0(I)]\Pr[\neg\mathcal{E}|\mathcal{Q}(I)]$ hold. By a simple calculation, we have

$$\begin{aligned}
|\Pr[F_1] - \tfrac{1}{2}| &= |\textstyle\sum_{I\in\mathcal{QID}} (\Pr[F_1(I) \wedge \mathcal{E}] + \Pr[F_1(I) \wedge \neg\mathcal{E}]) - \frac{1}{2}|\\
&= |\textstyle\sum_{I\in\mathcal{QID}} (\Pr[F_1(I) \wedge \neg\mathcal{E}] - \frac{1}{2}\Pr[\mathcal{Q}(I) \wedge \neg\mathcal{E}])|\\
&= |\textstyle\sum_{I\in\mathcal{QID}} (\Pr[F_0(I)] - \frac{1}{2}\Pr[\mathcal{Q}(I)])\Pr[\neg\mathcal{E}|\mathcal{Q}(I)]|.
\end{aligned}$$

Since $\Pr[F_i(I)] \le \Pr[Q(I)]$, we have $|\Pr[F_0(I) - \frac{1}{2}\Pr[\mathcal{Q}(I)]| \le \frac{1}{2}\Pr[\mathcal{Q}(I)]$ holds. Let $\eta(I) = \Pr[\neg\mathcal{E}|\mathcal{Q}(I)]$. Let $\eta_{max} = \max_{I\in\mathcal{QID}} \eta(I)$ and $\eta_{min} = \min_{I\in\mathcal{QID}} \eta(I)$. Then, we have

$$\begin{aligned}
|\Pr[F_1] - \tfrac{1}{2}| &= |\textstyle\sum_{I\in\mathcal{QID}} (\Pr[F_0(I)] - \frac{1}{2}\Pr[\mathcal{Q}(I)])(\eta_{min} + \eta(I) - \eta_{min})|\\
&\ge \eta_{min}|\textstyle\sum_{I\in\mathcal{QID}} (\Pr[F_0(I)] - \frac{1}{2}\Pr[\mathcal{Q}(I)])|\\
&\quad -|\textstyle\sum_{I\in\mathcal{QID}} (\Pr[F_0(I)] - \frac{1}{2}\Pr[\mathcal{Q}(I)])|(\eta(I) - \eta_{min})\\
&\ge \eta_{min}|\Pr[F_0] - \tfrac{1}{2}|\\
&\quad -\textstyle\sum_{I\in\mathcal{QID}} |\Pr[F_0(I)] - \frac{1}{2}\Pr[\mathcal{Q}(I)]|(\eta(I) - \eta_{min})\\
&\ge \eta_{min}|\Pr[F_0] - \tfrac{1}{2}| - \tfrac{1}{2}\textstyle\sum_{I\in\mathcal{QID}} \Pr[\mathcal{Q}(I)](\eta_{max} - \eta_{min})\\
&= \eta_{min}|\Pr[F_0] - \tfrac{1}{2}| - \tfrac{1}{2}(\eta_{max} - \eta_{min}).
\end{aligned}$$

Since $\mathcal{C}$ always samples $O(\epsilon^2 \log(\epsilon^{-1})\delta^{-1}\log(\delta^{-1}))$ times the probability p to compute p', we have that $\Pr[p' > p(1+\frac{\epsilon}{8})] < \delta\frac{\epsilon}{8}$ and $\Pr[p' < p(1-\frac{\epsilon}{8})] < \delta\frac{\epsilon}{8}$ hold by the Chernoff bounds. Since $p \ge \delta$, we have

$$\begin{aligned}
\eta_{max} &\le \max(\tfrac{\delta}{(1-\frac{\epsilon}{8})}, (1-\delta\tfrac{\epsilon}{8})p\tfrac{\delta}{p(1-\frac{\epsilon}{8})}) = \tfrac{\delta}{(1-\frac{\epsilon}{8})},\\
\eta_{min} &\ge (1-\delta\tfrac{\epsilon}{8})p\tfrac{\delta}{p(1+\frac{\epsilon}{8})} = (1-\delta\tfrac{\epsilon}{8})\tfrac{\delta}{(1+\frac{\epsilon}{8})}.
\end{aligned}$$

By Lemma 10, $|\Pr[F_0] - \frac{1}{2}| = \epsilon$ holds. Then, we have

$$\begin{aligned}
|\Pr[F_1] - \tfrac{1}{2}| &\ge \eta_{min}|\Pr[F_0] - \tfrac{1}{2}| - \tfrac{1}{2}(\eta_{max} - \eta_{min})\\
&\ge \tfrac{1}{2}\epsilon\delta.
\end{aligned}$$

This completes the proof of Lemma 11. □

Game 2. This game is identical to Game 1 except that the challenger $\mathcal{C}$ changes the setup phase as follows.

Setup. First compute $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^{\bar{n}}, 1^{\bar{m}}, q, \mathbf{I}_{\bar{n}})$ such that $\mathbf{A} \in \mathcal{R}_q^{\bar{n}\times\bar{m}}$, $\mathbf{R} = \mathcal{R}_q^{(\bar{m}-\bar{n}k)\times\bar{n}k}$. Then, randomly choose $\mathbf{U} \leftarrow \mathcal{R}_q^{\bar{n}\times\bar{n}}$, and compute $(K', td) \leftarrow \mathcal{H}.\mathrm{TrapGen}_{\mathcal{R}}(1^\kappa, \mathbf{A}, \mathbf{B})$. Finally, send $mpk = (\mathbf{A}, K', \mathbf{U})$ to the adversary $\mathcal{A}$, and keep the master secret key $\mathbf{R}$ and the trapdoor td private.

Lemma 12. *If $\mathcal{H}$ is a PHF, then* $|\Pr[F_2] - \Pr[F_1]| \leq \mathrm{negl}(\kappa)$.

Proof. By Definition 4, we have that the statistical distance between $(\mathbf{A}, K')$ and $(\mathbf{A}, K)$ is negligible for any $K \leftarrow \mathcal{H}.\mathrm{Gen}_{\mathcal{R}}(1^\kappa)$, $(K', td) \leftarrow \mathcal{H}.\mathrm{TrapGen}_{\mathcal{R}}(1^\kappa, \mathbf{A}, \mathbf{B})$. This means that the master public key mpk in Game 2 is statistically close to that in Game 1. Thus, we have $|\Pr[F_2] - \Pr[F_1]| \leq \mathrm{negl}(\kappa)$. □

Game 3. This game is identical to Game 2 except that the challenger $\mathcal{C}$ changes the way of generating the user private keys and the challenge ciphertext as follows.

Phase 1. Upon receiving the user private key query with identity $id \in \{0,1\}^\ell$, compute $\mathbf{A}_{id} = (\mathbf{A} \| \mathrm{H}_{K'}(id)) \in \mathcal{R}_q^{\bar{n}\times m}$, where $\mathrm{H}_{K'}(id) = \mathcal{H}.\mathrm{Eval}_{\mathcal{R}}(K', id) \in \mathcal{R}_q^{\bar{n}\times\bar{n}k}$. Then, compute $(\mathbf{R}_{id}, \mathbf{S}_{id}) = \mathcal{H}.\mathrm{TrapEval}_{\mathcal{R}}(td, K', id)$. If $\mathbf{S}_{id}$ is not invertible, the challenger $\mathcal{C}$ outputs a uniformly random bit and aborts the game. Otherwise, compute $\mathbf{E}_{id} \leftarrow \mathsf{SampleD}_{\mathcal{R}}(\mathbf{R}_{id}, \mathbf{A}_{id}, \mathbf{S}_{id}, \mathbf{U}, s)$, and send $sk_{id} = \mathbf{E}_{id} \in \mathcal{R}^{m\times n}$ to $\mathcal{A}$.

Challenge. This phase is the same as in Game 2 except that the challenger directly aborts and outputs a uniformly random bit if $\mathbf{S}_{id^*} \neq \mathbf{0}$, where $(\mathbf{R}_{id^*}, \mathbf{S}_{id^*}) = \mathcal{H}.\mathrm{TrapEval}_{\mathcal{R}}(td, K', id^*)$. Otherwise, it randomly chooses a bit $b^* \leftarrow \{0,1\}$, $\mathbf{s} \leftarrow \mathcal{R}_q^{\bar{n}}$, $\mathbf{x}_0 \leftarrow (D_{\mathcal{R},\alpha q})^{\bar{n}}, \mathbf{x}_1 \leftarrow (D_{\mathcal{R},\alpha q})^{\bar{m}}$. Then, it sets $\mathbf{C}_{b^*} = (\mathbf{c}_0^*, \mathbf{c}_1^* = (\mathbf{c}_{1,0}^*, \mathbf{c}_{1,1}^*))$, where

$$\mathbf{c}_0^* = \mathbf{U}^T\mathbf{s} + \mathbf{x}_0 + \frac{q}{2}M_{b^*}, \mathbf{c}_{1,0}^* = \mathbf{A}^T\mathbf{s} + \mathbf{x}_1, \mathbf{c}_{1,1}^* = \mathsf{ReRand}_{\mathcal{R}}((\mathbf{R}_{id^*})^T, \mathbf{c}_{1,0}^*, \alpha q, \beta).$$

Finally, it sends the challenge ciphertext $\mathbf{C}_{b^*}$ to the adversary $\mathcal{A}$.

Phase 2. $\mathcal{A}$ can adaptively make more user private key queries with any identity $id \neq id^*$. The challenger $\mathcal{C}$ responds as in Phase 1.

Lemma 13. *If $\mathcal{H}$ is a $(1, \mathsf{poly}, \beta, \gamma, \delta)$-PHF, then* $\Pr[F_3] = \Pr[F_2] - \mathrm{negl}(\kappa)$.

Proof. Note that both stages of the abort check and the artificial abort in Game 3 and Game 2 are identical. By the fact that the same abort conditions as in the abort check stage are examined when generating the user private keys and the ciphertext $\mathbf{C}_1 = (\mathbf{c}_0^*, \mathbf{c}_1^*)$, the challenger $\mathcal{C}$ in Game 3 will abort with the same probability as that in Game 2. Besides, if $\mathcal{C}$ does not abort in Game 3, we have that $\mathbf{S}_{id^*} = 0$ and $\mathbf{S}_{id}$ is invertible for any id in the user private key queries. In this case, $\mathcal{C}$ can use the $\mathsf{SampleD}_{\mathcal{R}}$ algorithm to successfully generate the user private keys by the fact that $s_1(\mathbf{R}_{id}) \leq \beta$ and $s > \max(\beta, \sqrt{m}) \cdot \omega(\sqrt{\log n\bar{n}})$. Moreover, if $\mathbf{S}_{id^*} = \mathbf{0}$, we have $\mathrm{H}_{K'}(id^*) = \mathbf{A}\mathbf{R}_{id^*}$. In this case, the challenge ciphertext $\mathbf{C}_{b^*} = (\mathbf{c}_0^*, \mathbf{c}_1^* = (\mathbf{c}_{1,0}^*, \mathbf{c}_{1,1}^*))$ in Game 2 has the following form:

$$\mathbf{c}_0^* = \mathbf{U}^T\mathbf{s} + \mathbf{x}_0 + \frac{q}{2}M_{b^*}, \mathbf{c}_{1,0}^* = \mathbf{A}^T\mathbf{s} + \mathbf{x}_1, \mathbf{c}_{1,1}^* = (\mathbf{R}_{id^*})^T(\mathbf{A}^T\mathbf{s}) + \mathbf{x}_2,$$

where $\mathbf{s} \leftarrow \mathcal{R}_q^{\bar{n}}$, $\mathbf{x}_0 \leftarrow (D_{\mathcal{R},\alpha q})^{\bar{n}}, \mathbf{x}_1 \leftarrow (D_{\mathcal{R},\alpha q})^{\bar{m}}$, and $\mathbf{x}_2 \leftarrow (D_{\mathcal{R},2\alpha\beta q})^{m'}$. Since $s_1(\mathbf{R}_{id^*}) \leq \beta$, we have that the challenge ciphertext $\mathbf{C}_1 = (\mathbf{c}_0^*, \mathbf{c}_1^* = (\mathbf{c}_{1,0}^*, \mathbf{c}_{1,1}^*))$ in Game 2 is statistically close to that in Game 3 by the property of $\mathsf{ReRand}_{\mathcal{R}}$.

Thus, if $\mathcal{C}$ does not abort during the game, then Game 3 is statistically close to Game 2 in the adversary $\mathcal{A}$'s view. In all, we have that $\Pr[F_3] = \Pr[F_2] - \mathrm{negl}(\kappa)$ holds. □

Game 4. This game is identical to Game 3 except that the challenger $\mathcal{C}$ changes the setup and the challenge phases as follows.

Setup. First randomly choose $\mathbf{A} \leftarrow \mathcal{R}_q^{\bar{n}\times\bar{m}}$, $\mathbf{U} \leftarrow \mathcal{R}_q^{\bar{n}\times\bar{n}}$, and compute $(K', td) \leftarrow \mathcal{H}.\mathrm{TrapGen}_{\mathcal{R}}(1^\kappa, \mathbf{A}, \mathbf{G})$. Then, send $mpk = (\mathbf{A}, K', \mathbf{U})$ to the adversary $\mathcal{A}$, and keep the trapdoor td private.

Challenge. This phase is the same as in Game 3 except that the challenger generates the ciphertext $\mathbf{C}_{b^*} = (\mathbf{c}_0^*, \mathbf{c}_1^* = (\mathbf{c}_{1,0}^*, \mathbf{c}_{1,1}^*))$ as follows: randomly choose vectors $\mathbf{b}_0 \leftarrow \mathcal{R}_q^{\bar{n}}, \mathbf{b}_1 \leftarrow \mathcal{R}_q^{\bar{m}}$, and compute

$$\mathbf{c}_0^* = \mathbf{b}_0 + \frac{q}{2}M_{b^*}, \mathbf{c}_{1,0}^* = \mathbf{b}_1, \mathbf{c}_{1,1}^* = \mathsf{ReRand}((\mathbf{R}_{id^*})^T, \mathbf{b}_1, \alpha q, \beta),$$

where $(\mathbf{R}_{id^*}, \mathbf{S}_{id^*}) = \mathcal{H}.\mathrm{TrapEval}(td, K', id^*)$.

Lemma 14. *If the advantage of any PPT algorithm $\mathcal{B}$ in solving the* $\mathrm{LWE}_{q,\alpha}$ *problem is at most ϵ', then we have that $|\Pr[F_4] - \Pr[F_3]| \leq \epsilon'$ holds.*

Proof. We construct an algorithm $\mathcal{B}$ for the $\mathrm{LWE}_{q,\alpha}$ over $\mathcal{R}$ as follows. Given the $\mathrm{LWE}_{q,\alpha}$ challenge instance $(\hat{\mathbf{U}}, \hat{\mathbf{b}}_0) \in \mathcal{R}_q^{\bar{n}\times\bar{n}} \times \mathcal{R}_q^{\bar{n}}$ and $(\hat{\mathbf{A}}, \hat{\mathbf{b}}_1) \in \mathcal{R}_q^{\bar{n}\times\bar{m}} \times \mathcal{R}_q^{\bar{m}}$. $\mathcal{B}$ simulates the security game for the adversary $\mathcal{A}$ the same as in Game 3 except that it replaces $(\mathbf{A}, \mathbf{U})$ in the setup phase and $(\mathbf{b}_0, \mathbf{b}_1)$ in the challenge phase with $(\hat{\mathbf{A}}, \hat{\mathbf{U}})$ and $(\hat{\mathbf{b}}_0, \hat{\mathbf{b}}_1)$, respectively.

It is easy to check that if $(\hat{\mathbf{U}}, \hat{\mathbf{b}}_0) \in \mathcal{R}_q^{\bar{n}\times\bar{n}} \times \mathcal{R}_q^{\bar{n}}$ and $(\hat{\mathbf{A}}, \hat{\mathbf{b}}_1) \in \mathcal{R}_q^{\bar{n}\times\bar{m}} \times \mathcal{R}_q^{\bar{m}}$ are valid LWE tuples, then $\mathcal{A}$ is in Game 3, otherwise $\mathcal{A}$ is in Game 4. This means that $\mathcal{B}$ is a valid LWE distinguisher, which implies that both $|\Pr[F_4]-\Pr[F_3]| \leq \epsilon'$ holds by our assumption. □

Lemma 15. $\Pr[F_4] = \frac{1}{2}$.

Proof. The claim follows from the fact that $\mathbf{b}_0$ is uniformly random. □

By Lemma 11∼13, we have $|\Pr[F_3] - \frac{1}{2}| \geq \frac{1}{2}\epsilon\delta - \text{negl}(\kappa)$. By Lemma 15, we have $\Pr[F_4] = \frac{1}{2}$. By the fact that $|\Pr[F_4] - \Pr[F_3]| \leq \epsilon'$ in Lemma 14, we have $\epsilon' \geq \frac{\epsilon\delta}{2} - \text{negl}(\kappa)$ holds. This completes the proof of the first claim in Theorem 12. The second claim directly follows from the fact that the adversary will only make at most $Q \leq v$ user private key queries. □

6.3 Instantiations

By instantiating the generic IBE scheme $\mathcal{IBE}$ with our Type-II $(1, v, \beta)$-PHF in Definition 6, we can obtain an IBE scheme with master public key containing $O(\log n)$ number of matrices. Let $\mathcal{IBE}_1$ be the instantiated scheme.

Corollary 3. *If there exists a PPT adversary $\mathcal{A}$ breaking the* IND-qID-CPA *security of $\mathcal{IBE}_1$ with non-negligible advantage ϵ and making at most $Q \leq v$ user private key queries, then there exists an algorithm $\mathcal{B}$ solving the* $\text{LWE}_{q,\alpha}$ *problem over $\mathcal{R} = \mathbb{Z}$ with advantage at least $\epsilon' \geq \frac{\epsilon}{32nv^2} - \text{negl}(\kappa)$.*

Similarly, by instantiating the generic IBE scheme $\mathcal{IBE}$ with our Type-III $(1, v, \beta)$-PHF with $\mathcal{R} = R$ in Definition 8, we can obtain an IBE scheme with master public key containing constant number of matrices. Let $\mathcal{IBE}_2$ be the instantiated scheme.

Corollary 4. *If there exists a PPT adversary $\mathcal{A}$ breaking the* IND-qID-CPA *security of $\mathcal{IBE}_2$ with non-negligible advantage ϵ and making at most $Q \leq v$ user private key queries, then there exists an algorithm $\mathcal{B}$ solving the* $\text{LWE}_{q,\alpha}$ *problem over ring $\mathcal{R} = R$ with advantage at least $\epsilon' \geq \frac{\epsilon}{32nv^2} - \text{negl}(\kappa)$.*

6.4 Extensions

Hierarchical IBE. Using the trapdoor delegation techniques in [2,16,47], one can extend our generic IBE scheme $\mathcal{IBE}$ into a generic hierarchical IBE (HIBE) scheme. We now give a sketch of the construction. For identity depth $d \geq 1$, we include d different PHF keys $\{K_i\}_{i\in\{1,\ldots,d\}}$ in the master public key, and the "public key" $\mathbf{A}_{id}$ for any identity $id = (id_1, \ldots, id_{d'})$ with depth $d' \leq d$ is defined as $\mathbf{A}_{id} = (\mathbf{A}\|\text{H}_{K_1}(id_1)\|\cdots\|\text{H}_{K_{d'}}(id_{d'}))$. Then, one can use $\mathbf{A}_{id}$ to encrypt plaintexts the same as in our generic IBE scheme. In order to enable the delegation of user private keys, the user private key should be replaced by a new trapdoor extended by the trapdoor of $\mathbf{A}$ using the algorithms in [2,16,47]. We note that as previous schemes using similar partitioning techniques [2,16], such a construction seems to inherently suffer from a reduction loss depending on the identity depth d in the exponent. It is still unclear whether one can adapt the dual system of Waters [55] to construct lattice-based (H)IBEs with tight security proofs.

Chosen Ciphertexts Security. Obviously, one can use the CHK technique in [15] to transform a CPA secure HIBE for identity depth d to a CCA secure HIBE for identity depth $d - 1$, by appending each identity in the encryption with the verification key of a one-time strongly EUF-CMA signature scheme. In our case, one can obtain an IND-ID-CCA secure IBE scheme by using a two-level IND-ID-CPA HIBE scheme. Since the CHK technique only requires "selective-security" to deal with the one-time signature's verification key, we can construct a more efficient CCA secure IBE scheme by directly combining a normal PHF with a weak one. Since a weak PHF is usually simpler and more efficient, the resulting IBE could be more efficient than the one obtained by directly applying the CHK technique to a two-level fully secure HIBE scheme. We now give the sketch of the improved construction. In addition to a normal PHF key K in the master public key of our generic IBE scheme $\mathcal{IBE}$, we also include it a weak PHF key K_1. When generating user private key for identity id, we compute a new trapdoor of $\mathbf{A}_{id} = (\mathbf{A} \| \mathrm{H}_K(id))$ as the user private key, by using the trapdoor delegation algorithms in [2,16,47]. In the encryption algorithm, we generate a one-time signature verification key vk (for simplicity we assume the length of vk is compatible with the weak PHF), and uses the matrix $\mathbf{A}_{id,vk} = (\mathbf{A}_{id} \| \mathrm{H}_{K_1}(vk)) = (\mathbf{A} \| \mathrm{H}_K(id) \| \mathrm{H}_{K_1}(vk))$ to encrypt the plaintext as $\mathcal{IBE}.\mathsf{Enc}$. The decryption algorithm is the same as $\mathcal{IBE}.\mathsf{Dec}$ except that it first computes the "user private key" for $\mathbf{A}_{id,vk}$ from the user private key of $\mathbf{A}_{id}$.

7 Conclusions and Open Problems

We proposed the notion of lattice-based PHFs and three types of concrete constructions. We showed that under the ISIS assumption, any non-trivial lattice-based PHFs imply a collision-resistant hash function. We provided a generic signature scheme from lattice-based PHFs, which encompassed the lattice-based signature schemes in [13,47]. By instantiating the generic scheme with our efficient lattice-based PHF constructions, we immediately obtained lattice-based short signature schemes with short verification keys. Furthermore, we showed how to combine different lattice-based PHFs to construct short signature schemes from weaker assumptions. We also constructed a generic construction of IBE scheme from lattice-based PHFs. By instantiating the generic IBE scheme with our efficient lattice-based PHF constructions, we obtained IBE scheme with short master public keys in the standard model. We also showed how to extend our (generic) IBE scheme into a (generic) HIBE scheme and how to achieve CCA security.

One interesting problem is to give a simpler formalization of PHFs that captures both the DL setting and the lattice setting. Another interesting problem is to find more (efficient) constructions/applications of lattice-based PHFs.

Acknowledgments. We thank the anonymous reviewers for helpful comments on earlier versions of our paper. Jiang Zhang, the corresponding author, is supported by the National Natural Science Foundation of China (Grant Nos. 62022018, 61932019) and the National Key Research and Development Program of China (Grant No. 2022YFB2702000). Yu Chen is supported by the National Key Research and Development Program of China (Grant No. 2021YFA1000600), the National Natural Science Foundation of China (Grant Nos. 62272269, 61932019), and Taishan Scholar Program of Shandong Province. Zhenfeng Zhang is supported by the National Key Research and Development Program of China (No. 2022YFB2701600).

References

1. Abla, P., Liu, F.H., Wang, H., Wang, Z.: Ring-based identity based encryption – asymptotically shorter mpk and tighter security. In: Nissim, K., Waters, B. (eds.) Theory of Cryptography. pp. 157–187. Springer International Publishing, Cham (2021)
2. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: Gilbert, H. (ed.) EUROCRYPT 2010, LNCS, vol. 6110, pp. 553–572. Springer (2010)
3. Agrawal, S., Boneh, D., Boyen, X.: Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In: Rabin, T. (ed.) CRYPTO 2010, LNCS, vol. 6223, pp. 98–115. Springer (2010)
4. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: STOC '96. pp. 99–108. ACM (1996)
5. Ajtai, M.: Generating hard instances of the short basis problem. In: Wiedermann, J., van Emde Boas, P., Nielsen, M. (eds.) Automata, Languages and Programming, LNCS, vol. 1644, pp. 706–706. Springer (1999)
6. Alperin-Sheriff, J.: Short signatures with short public keys from homomorphic trapdoor functions. In: Katz, J. (ed.) PKC 2015, LNCS, vol. 9020, pp. 236–255. Springer (2015)
7. Bai, S., Galbraith, S.: An improved compression technique for signatures based on learning with errors. In: Benaloh, J. (ed.) CT-RSA 2014, LNCS, vol. 8366, pp. 28–47. Springer International Publishing (2014)
8. Bellare, M., Ristenpart, T.: Simulation without the artificial abort: Simplified proof and improved concrete security for Waters' IBE scheme. In: Joux, A. (ed.) EUROCRYPT 2009, LNCS, vol. 5479, pp. 407–424. Springer (2009)
9. Böhl, F., Hofheinz, D., Jager, T., Koch, J., Seo, J., Striecks, C.: Practical signatures from standard assumptions. In: Johansson, T., Nguyen, P. (eds.) EUROCRYPT 2013, LNCS, vol. 7881, pp. 461–485. Springer (2013)
10. Boneh, D., Boyen, X.: Efficient selective-ID secure identity-based encryption without random oracles. In: Cachin, C., Camenisch, J. (eds.) Advances in Cryptology – EUROCRYPT 2004, LNCS, vol. 3027, pp. 223–238. Springer (2004)

11. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. In: Kilian, J. (ed.) CRYPTO 2001, LNCS, vol. 2139, pp. 213–229. Springer (2001)
12. Boneh, D., Venkatesan, R.: Breaking RSA may not be equivalent to factoring. In: Nyberg, K. (ed.) EUROCRYPT '98, LNCS, vol. 1403, pp. 59–71. Springer (1998)
13. Boyen, X.: Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In: Nguyen, P., Pointcheval, D. (eds.) PKC 2010, LNCS, vol. 6056, pp. 499–517. Springer (2010)
14. Boyen, X., Li, Q.: Towards tightly secure lattice short signature and id-based encryption. In: ASIACRYPT 2016. pp. 404–434. Springer (2016)
15. Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004, LNCS, vol. 3027, pp. 207–222. Springer (2004)
16. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In: Gilbert, H. (ed.) EUROCRYPT 2010, LNCS, vol. 6110, pp. 523–552. Springer (2010)
17. Catalano, D., Fiore, D., Nizzardo, L.: Programmable hash functions go private: Constructions and applications to (homomorphic) signatures with shorter public keys. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015, LNCS, vol. 9216, pp. 254–274. Springer (2015)
18. Cheon, J., Han, K., Lee, C., Ryu, H., Stehlé, D.: Cryptanalysis of the multilinear map over the integers. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, LNCS, vol. 9056, pp. 3–12. Springer (2015)
19. Cocks, C.: An identity based encryption scheme based on quadratic residues. In: Honary, B. (ed.) Cryptography and Coding, LNCS, vol. 2260, pp. 360–363. Springer (2001)
20. Coron, J.S., Gentry, C., Halevi, S., Lepoint, T., Maji, H., Miles, E., Raykova, M., Sahai, A., Tibouchi, M.: Zeroizing without low-level zeroes: New MMAP attacks and their limitations. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015, LNCS, vol. 9215, pp. 247–266. Springer (2015)
21. Coron, J.S., Lepoint, T., Tibouchi, M.: Practical multilinear maps over the integers. In: Canetti, R., Garay, J. (eds.) CRYPTO 2013, LNCS, vol. 8042, pp. 476–493. Springer (2013)
22. Cramer, R., Damgård, I.: On the amortized complexity of zero-knowledge protocols. In: Halevi, S. (ed.) CRYPTO 2009, LNCS, vol. 5677, pp. 177–191. Springer (2009)
23. Dodis, Y., Rafail, O., Reyzin, L., Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. SIAM Journal on Computing 38, 97–139 (2008)
24. Ducas, L., Durmus, A., Lepoint, T., Lyubashevsky, V.: Lattice signatures and bimodal gaussians. In: Canetti, R., Garay, J. (eds.) CRYPTO 2013, LNCS, vol. 8042, pp. 40–56. Springer (2013)
25. Ducas, L., Lyubashevsky, V., Prest, T.: Efficient identity-based encryption over NTRU lattices. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, LNCS, vol. 8874, pp. 22–41. Springer (2014)
26. Ducas, L., Micciancio, D.: Improved short lattice signatures in the standard model. In: Garay, J., Gennaro, R. (eds.) CRYPTO 2014, LNCS, vol. 8616, pp. 335–352. Springer (2014)
27. Erdös, P., Frankl, P., Füredi, Z.: Families of finite sets in which no set is covered by the union of r others. Israel Journal of Mathematics 51(1-2), 79–89 (1985)

28. Freire, E., Hofheinz, D., Paterson, K., Striecks, C.: Programmable hash functions in the multilinear setting. In: Canetti, R., Garay, J. (eds.) CRYPTO 2013, LNCS, vol. 8042, pp. 513–530. Springer (2013)
29. Garg, S., Gentry, C., Halevi, S.: Candidate multilinear maps from ideal lattices. In: Johansson, T., Nguyen, P. (eds.) EUROCRYPT 2013, LNCS, vol. 7881, pp. 1–17. Springer (2013)
30. Gentry, C.: Practical identity-based encryption without random oracles. In: Vaudenay, S. (ed.) Advances in Cryptology – EUROCRYPT 2006, LNCS, vol. 4004, pp. 445–464. Springer (2006)
31. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC 2008. pp. 197–206. ACM (2008)
32. Gorbunov, S., Vaikuntanathan, V., Wichs, D.: Leveled fully homomorphic signatures from standard lattices. In: STOC 2015. pp. 469–477. ACM (2015)
33. Hanaoka, G., Matsuda, T., Schuldt, J.: On the impossibility of constructing efficient key encapsulation and programmable hash functions in prime order groups. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012, LNCS, vol. 7417, pp. 812–831. Springer (2012)
34. Hofheinz, D., Jager, T., Kiltz, E.: Short signatures from weaker assumptions. In: Lee, D., Wang, X. (eds.) ASIACRYPT 2011, LNCS, vol. 7073, pp. 647–666. Springer (2011)
35. Hofheinz, D., Kiltz, E.: Programmable hash functions and their applications. In: Wagner, D. (ed.) CRYPTO 2008, LNCS, vol. 5157, pp. 21–38. Springer (2008)
36. Hofheinz, D., Kiltz, E.: Programmable hash functions and their applications. Journal of Cryptology 25(3), 484–527 (2012)
37. Hu, Y., Jia, H.: Cryptanalysis of GGH map. In: Fischlin, M., Coron, J.S. (eds.) EUROCRYPT 2016, LNCS, vol. 9665, pp. 537–565. Springer (2016)
38. Jager, T., Kurek, R., Niehues, D.: Efficient adaptively-secure ib-kems and vrfs via near-collision resistance. In: Garay, J.A. (ed.) Public-Key Cryptography – PKC 2021. pp. 596–626. Springer International Publishing, Cham (2021)
39. Kajita, K., Ogawa, K., Nuida, K., Takagi, T.: Short lattice signatures in the standard model with efficient tag generation. In: Nguyen, K., Wu, W., Lam, K.Y., Wang, H. (eds.) Provable and Practical Security. pp. 85–102. Springer International Publishing, Cham (2020)
40. Katsumata, S., Yamada, S.: Partitioning via non-linear polynomial functions: more compact ibes from ideal lattices and bilinear maps. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 682–712. Springer (2016)
41. Katz, J.: Digital Signatures. Springer (2010)
42. Krawczyk, H., Rabin, T.: Chameleon signatures. In: NDSS 2000.
43. Kumar, R., Rajagopalan, S., Sahai, A.: Coding constructions for blacklisting problems without computational assumptions. In: CRYPTO '99. pp. 609–623. Springer (1999)
44. Lyubashevsky, V.: Lattice signatures without trapdoors. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012, LNCS, vol. 7237, pp. 738–755. Springer (2012)
45. Lyubashevsky, V., Micciancio, D.: Asymptotically efficient lattice-based digital signatures. In: Canetti, R. (ed.) Theory of Cryptography, LNCS, vol. 4948, pp. 37–54. Springer (2008)
46. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Gilbert, H. (ed.) EUROCRYPT 2010, LNCS, vol. 6110, pp. 1–23. Springer (2010)

47. Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012, LNCS, vol. 7237, pp. 700–718. Springer (2012)
48. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on gaussian measures. SIAM J. Comput. 37, 267–302 (April 2007)
49. Nguyen, P., Zhang, J., Zhang, Z.: Simpler efficient group signatures from lattices. In: Katz, J. (ed.) PKC 2015, LNCS, vol. 9020, pp. 401–426. Springer (2015)
50. Peikert, C., Rosen, A.: Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In: Halevi, S., Rabin, T. (eds.) Theory of Cryptography, LNCS, vol. 3876, pp. 145–166. Springer (2006)
51. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC 2005. pp. 84–93. ACM (2005)
52. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G., Chaum, D. (eds.) CRYPTO '84, LNCS, vol. 196, pp. 47–53. Springer (1984)
53. Vershynin, R.: Introduction to the non-asymptotic analysis of random matrices. arXiv preprint arXiv:1011.3027 (2010)
54. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005, LNCS, vol. 3494, pp. 114–127. Springer (2005)
55. Waters, B.: Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In: Halevi, S. (ed.) CRYPTO 2009, LNCS, vol. 5677, pp. 619–636. Springer (2009)
56. Yamada, S.: Adaptively secure identity-based encryption from lattices with asymptotically shorter public parameters. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 32–62. Springer (2016)
57. Yamada, S.: Asymptotically compact adaptively secure lattice ibes and verifiable random functions via generalized partitioning techniques. In: Annual International Cryptology Conference. pp. 161–193. Springer (2017)
58. Yamada, S., Hanaoka, G., Kunihiro, N.: Two-dimensional representation of cover free families and its applications: Short signatures and more. In: Dunkelman, O. (ed.) CT-RSA 2012, LNCS, vol. 7178, pp. 260–277. Springer (2012)
59. Zhandry, M.: Secure identity-based encryption in the quantum random oracle model. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012, LNCS, vol. 7417, pp. 643–662. Springer (2012)
60. Zhang, J., Chen, Y., Zhang, Z.: Programmable hash functions from lattices: Short signatures and IBEs with small key sizes. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, LNCS, vol. 9816, pp. 303–332. Springer, Heidelberg (2016)