Programmable Hash Functions from Lattices: Short Signatures and IBEs with Small Key Sizes

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

[1] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China
[2] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, 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 two types of 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 collision-resistant, 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 a new short signature scheme and a new fully secure IBE scheme with keys consisting of a logarithmic number of matrices/vectors in the security parameter κ. Besides, we also give a refined way of combining two concrete PHFs to construct an improved short signature scheme with short verification keys from weaker assumptions. 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 existential unforgeability against chosen message attacks (EUF-CMA) without resorting to chameleon hash functions.

1 Introduction

As a primitive catpuring the partinitioning 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 $\Pr[a_{X_1} = \cdots = a_{X_u} = 0 \wedge a_{Y_1}, \ldots, a_{Y_v} \neq 0] \geq 1/\mathrm{poly}(\kappa)$ for some polynomial $\mathrm{poly}(\kappa)$ 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 [55,33,36,28,17], 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 [38] 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 [42,31]. Later, many works (e.g., [41,24,8]) 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 Crypto 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 [14], which was later improved by Micciancio and Peikert [45]. However, the verification keys of both schemes in [14,45] consist of a linear number of matrices.

In 2013, Böhl et al. [10] 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. Recently, 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 [10,26,6] 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 [39], in the lattice setting [26] this usually introduces an additional tag to each signature and roughly increases the signature size by twice. 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 [50] introduced identity-based encryption (IBE) in 1984, but the first realizations were due to Boneh and Franklin from pairings [12] 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 [2,16,56,25] 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 [1,2,16]. Concretely, the scheme in [2] was only proven to be *selective-identity* secure in the standard model. By using standard complexity leverage technique [11], one can generally transform a selective-identity secure IBE scheme into a *full*

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 [1,16], 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 [1,16] 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 [1].

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 several different 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. Under the ISIS assumption, our generic signature scheme is EUF-CMA secure in the standard model. We also give a generic IBE scheme from lattice-based PHFs with a property called high min-entropy. Under the LWE assumption, our generic IBE scheme is secure against adaptive chosen-plaintext and chosen-identity attacks in the standard model. 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 [14,45] and the IBE schemes [1]. 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 new PHF constructions, we obtain a new short signature scheme and a new IBE scheme. Compared to previous schemes, our instantiated schemes have several appealing advantages. Besides, we also construct an improved short signature scheme with short verification keys by carefully combining two concrete PHFs. Comparisons between our schemes and previous ones will be given in Section 1.3 and Section 1.4.

1.2 Techniques

We introduce the notion of lattice-based PHFs by carefully exploiting the specific algebraic structure of lattices. 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 $\mathrm{H}_K(X) \in \mathbb{Z}_q^{n\times m}$ for some positive $n, m, q \in \mathbb{Z}$. In the trapdoor mode, there additionally exists a secret trapdoor td allowing to compute matrices $\mathbf{R}_X \in \mathbb{Z}_q^{\bar{m}\times m}$ and $\mathbf{S}_X \in \mathbb{Z}_q^{n\times n}$ for some integer $\bar{m} \in \mathbb{Z}$, such that $\mathrm{H}_K(X) = \mathbf{A}\mathbf{R}_X + \mathbf{S}_X\mathbf{B} \in \mathbb{Z}_q^{n\times m}$ holds with respect to user-specified "generators" $\mathbf{A} \in \mathbb{Z}_q^{n\times\bar{m}}$ and $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$. 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" $\mathbf{A} \in \mathbb{Z}_q^{n\times\bar{m}}$ and $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$ have essential differences

for embedding hard lattice problems. More precisely, in our definition $\mathbf{A} \in \mathbb{Z}_q^{n \times \bar{m}}$ is required to be uniformly distributed (and thus can be used to embed the ISIS problem), while $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$ is a trapdoor matrix that allows to efficiently sample short vector $\mathbf{e} \in \mathbb{Z}^m$ satisfying $\mathbf{Be} = \mathbf{v}$ for any vector $\mathbf{v} \in \mathbb{Z}_q^n$.

In order to explore the differences between $\mathbf{A} \in \mathbb{Z}_q^{n \times \bar{m}}$ and $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$ in the security reduction, we require that the largest singular value of $\mathbf{R}_X$ defined by $s_1(\mathbf{R}_X) = \max_{\mathbf{u}} \|\mathbf{R}_X \mathbf{u}\|$ is small where the maximum is taken over all unit vectors $\mathbf{u} \in \mathbb{R}^m$, and that $\mathbf{S}_X \in \mathcal{I}_n \cup \{\mathbf{0}\}$ where $\mathcal{I}_n$ is the set of invertible matrices in $\mathbb{Z}_q^{n \times n}$. 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 $1/\mathrm{poly}(n)$, 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 $\mathbf{A} \in \mathbb{Z}_q^{n \times \bar{m}}$ and trapdoor matrix $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$, then for any input X the matrix $\mathbf{A}_X := (\mathbf{A} \| \mathrm{H}_{K'}(X)) = (\mathbf{A} \| \mathbf{AR}_X + \mathbf{S}_X \mathbf{B}) \in \mathbb{Z}_q^{n \times (\bar{m}+m)}$ 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 $\mathbf{v} \in \mathbb{Z}_q^n$ 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" $\mathbf{A} \in \mathbb{Z}_q^{n \times \bar{m}}$, we require the keys in both modes to be statistically indistinguishable even conditioned on the information of $\mathbf{A}$.

Our Type-I PHF construction is a high-level abstraction of the functions that were (implicitly) used in both signature schemes (e.g, [14,10,45]) and encryption schemes (e.g., [1,45]). Formally, let E be an encoding function from some domain $\mathcal{X}$ to $(\mathbb{Z}_q^{n \times n})^\ell$, 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 literatures (implicitly) showed that the Type-I construction satisfies our definition of lattice-based PHFs. Concretely, if one sets $\mathcal{X} = \{0,1\}^\ell$, 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 [14] and the IBE scheme in [1]. Since the Type-I PHF construction is independent from the particular choice of $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$, it allows us to use any trapdoor matrix $\mathbf{B}$ when generating the trapdoor mode key. On the downside, such a construction has a large key size, i.e., the number of matrices in the key is linear in the input length ℓ.

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 [45], where $\mathbf{G} = \mathbf{I}_n \otimes \mathbf{g}^t \in \mathbb{Z}_q^{n \times nk}$, $k = \lceil \log_2 q \rceil$ and $\mathbf{g} = (1, 2, \ldots, 2^{k-1})^t \in \mathbb{Z}_q^k$. 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 N is as large as $\mathrm{poly}(\ell)$. We solve this problem by leveraging 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 any input $X \in [L]$, 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

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

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 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 $1/N = 1/\mathrm{poly}(\ell)$.

1.3 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 $\mathbb{Z}_q^{n\times m'}$, let $\mathbf{B} \in \mathbb{Z}_q^{n\times m'}$ 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 $\mathbf{A} \in \mathbb{Z}_q^{n\times\bar{m}}$, a uniformly random vector $\mathbf{u} \in \mathbb{Z}_q^n$, 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 $\mathbf{v} \in \mathbb{Z}_q^n$. Given a message $M \in \{0,1\}^\ell$, the signing algorithm first computes $\mathbf{A}_M = (\mathbf{A} \| \mathrm{H}_K(M)) \in \mathbb{Z}_q^{n\times m}$, and then uses the trapdoor $\mathbf{R}$ to sample a short vector $\mathbf{e} \in \mathbb{Z}^m$ satisfying $\mathbf{A}_M\mathbf{e} = \mathbf{u}$ by employing the sampling algorithms in [31,16,45]. 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 [31,45].

In addition, if $\mathcal{H} = \{\mathrm{H}_K\}$ is a $(1, v, \beta)$-PHF for some integer v and real β, we can show that under the ISIS assumption, $\mathcal{SIG}$ is existentially unforgeable against adaptive chosen message attacks (EUF-CMA) in the standard model as long as the forger $\mathcal{F}$ makes at most $Q \leq v$ 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 $\mathbf{A}_{M_i} = (\hat{\mathbf{A}} \| \mathrm{H}_{K'}(M_i)) = (\hat{\mathbf{A}} \| \hat{\mathbf{A}}\mathbf{R}_{M_i} + \mathbf{S}_{M_i}\mathbf{B}) \in \mathbb{Z}_q^{n\times m}$. 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 $\{M_i\}_{i\in\{1,\ldots,Q\}}$, 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 $\mathbf{u} = \mathbf{A}_{M^*}\sigma^* = \hat{\mathbf{A}}(\mathbf{I}_{\bar{m}} \| \mathbf{R}_{M^*})\sigma^*$.

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 "CMH?" denotes the necessity of chameleon hash functions to achieve EUF-CMA security. Constant $c > 1$ and $d = O(\log_c n)$ are the parameters in [10,26,6])

Schemes	Verification key	Signature	Reduction loss	(I)SIS param $\bar{\beta}$	CMH?
LM08 [42] *	1	$\log n$	Q	$\tilde{O}(n^2)$	NO
CHKP10 [16]	n	$\log n$	Q	$\tilde{O}(n^{1.5})$	YES
Boyen10 [14]	n	1	Q	$\tilde{O}(n^{3.5})$	NO
MP12 [45] †	n	1	Q	$\tilde{O}(n^{2.5})$	YES
BHJ$^+$14 [10]	1	d	$(Q^2/\epsilon)^c$	$\tilde{O}(n^{2.5})$	YES
DM14 [26] *	d	1	$(Q^2/\epsilon)^c$	$\tilde{O}(n^{3.5})$	YES
AS15 [6]	1	1	$(Q^2/\epsilon)^c$	$\tilde{O}(d^{2d} \cdot n^{5.5})$	YES
Our $\mathcal{SIG}_1$	$\log n$	1	$n \cdot Q^2$	$Q^2 \cdot \tilde{O}(n^{5.5})$	NO
Our $\mathcal{SIG}_2$	$\log n$	1	$Q \cdot \tilde{O}(n)$	$\tilde{O}(n^{5.5})$	NO

Each signature in the generic scheme $\mathcal{SIG}$ only has a single vector, which is as short as that in [14,45]. In fact, our generic scheme $\mathcal{SIG}$ encompasses the two signature schemes from [14,45] 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 [14,45] had large verification keys consisting of a linear number of matrices. By instantiating $\mathcal{SIG}$ with our efficient Type-II PHF construction, we obtain a concrete scheme $\mathcal{SIG}_1$ with verification keys consisting of a logarithmic number of matrices. Unlike the prior schemes in [10,26,6], our methods do not use the confined guessing proof technique [10], and enable us to directly achieve EUF-CMA security without using chameleon hash functions. This also allows us to get a security proof of $\mathcal{SIG}_1$ with a reduction loss only about nv^2, which is independent from the forger's success probability ϵ. We remark that this improvement does not come for free: the underlying ISIS assumption should hold for parameter $\bar{\beta} = v^2 \cdot \tilde{O}(n^{5.5})$, where $v \geq Q$ is required.[4] By carefully combining our Type-II $(1, v, \beta)$-PHF with a simple weak Type-I PHF and introducing a very short tag to each signature, we further remove the condition $v \geq Q$ such that a much smaller $v = \omega(\log n)$ can be used to construct an improved short signature scheme $\mathcal{SIG}_2$ from (relatively) weaker ISIS assumption, which further removes a factor of Q^2 (resp. Q) from the ISIS parameter (resp. the reduction loss) for our generic signature scheme.

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 [10,26,6]. 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 [16,14,45,10,6] and ours can be instantiated from ideal lattices, and thus roughly saves a factor of n in the

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

verification key size. However, the two schemes [42,26] (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 [45] (marked with '†') is essentially identical to the one in [14] except that an improved security reduction under a weaker assumption was provided in the EUF–naCMA model. 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 schemes. 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 schemes.

1.4 Identity-based Encryptions

At STOC 2008, Gentry et al. [31] constructed a variant of the LWE-based public-key encryption (PKE) scheme [49]. 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 $\mathbf{A}_M = (\mathbf{A} \| \mathrm{H}_K(M))$. A natural question is whether we can construct a generic IBE scheme 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 algorithm of [31] with "public key" $(\mathbf{A}_{id}, \mathbf{u})$ as a sub-routine to encrypt plaintexts under the identity id. The problem is that we do not know how to rely the security of the above "IBE" scheme on the LWE assumption.

Fortunately, the work [1] suggested a solution by adding an "artificial" noise in the ciphertext, which was later used in other advanced lattice-based encryption schemes such as functional encryptions [3]. To adapt their techniques to the above IBE construction, the challenge ciphertext $\mathbf{C}^*$ under identity id^* must contain a term $\mathbf{R}_{id^*}^t \mathbf{w}$ for some $\mathbf{w} \in \mathbb{Z}_q^{\bar{m}}$, where $\mathrm{H}_{K'}(id^*) = \mathbf{A}\mathbf{R}_{id^*}$ (i.e., $\mathbf{S}_{id^*} = \mathbf{0}$) for some trapdoor mode key K'. This means that $\mathbf{C}^*$ will leak some information of $\mathbf{R}_{id^*}$, which is not captured by our definition of lattice-based PHF, and might compromise the security of $\mathcal{H}$. An intuitive solution is directly resorting to an enhanced definition of PHF such that all the properties of $\mathcal{H}$ still hold even when the information of $\mathbf{R}_{id^*}^t \mathbf{w}$ (for any given $\mathbf{w}$) is leaked. For our particular generic construction of IBE, we can handle it more skillfully by introducing two seemingly relaxed conditions: 1) the PHF key K' in the trapdoor mode is still statistically close to the key K in the normal mode even conditioned on ($\mathbf{A}$ and) $\mathbf{R}_{id^*}^t \mathbf{w}$ for any given vector $\mathbf{w} \in \mathbb{Z}_q^{\bar{m}}$; 2) the hidden matrix $\mathbf{R}_{id^*}$ has high min-entropy in the sense that $\mathbf{R}_{id^*}^t \mathbf{w}$ (conditioned on $\mathbf{w}$) is statistically close to uniform over $\mathbb{Z}_q^m$ when $\mathbf{w} \in \mathbb{Z}_q^{\bar{m}}$ is uniformly random. Formally, we say that a PHF $\mathcal{H}$ has *high min-entropy* if it additionally satisfies the above two conditions. Intuitively, the high min-entropy property ensures that when $\mathbf{w}$ is uniformly random, $\mathbf{R}_{id^*}^t \mathbf{w}$ statistically leaks no information of $\mathbf{R}_{id^*}$, and thus will not affect the original PHF property of $\mathcal{H}$. In the security proof, we will make use of this fact by switching $\mathbf{w}$ to a uniformly random one under the LWE assumption. Interestingly, by choosing appropriate parameters, all our PHF constructions satisfy the high min-entropy property. In other words, such a property is obtained almost for free, which finally allows us to construct a generic IBE scheme $\mathcal{IBE}$ from lattice-based PHFs with high min-entropy. Similarly, our generic scheme $\mathcal{IBE}$ subsumes the concrete IBE schemes due to Agrawal et al. [1].

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)

Schemes	Master public key	Ciphertext	Reduction loss	LWE param $1/\alpha$	Security
ABB10a [2]	n^3	n^2	1	$\tilde{O}(n^{2n})$	Selective
ABB10b [1]	$1, n$	1	$1, Q$	$\tilde{O}(n^2)$	Selective, Full
CHKP10 [16]	n	n	Q^2	$\tilde{O}(n^{1.5})$	Full
Our $\mathcal{IBE}_1$	$\log n$	1	$n \cdot Q^2$	$Q^2 \cdot \tilde{O}(n^{6.5})$	Full

Besides, by instantiating $\mathcal{IBE}$ with our efficient Type-II PHF construction, we obtain the first standard model IBE scheme $\mathcal{IBE}_1$ with master public keys consisting of a logarithmic number of matrices. We also show how to extend our IBE scheme to a hierarchical IBE (HIBE) scheme and how to achieve CCA security, by using the trapdoor delegations [1,16,45] and the CHK transformation [15].

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 the two fully secure IBEs [1,16] in the standard model, our concrete scheme $\mathcal{IBE}_1$ only has a logarithmic number of matrices in the master public key. However, such an improvement is not obtained without a penalty: the instantiated scheme $\mathcal{IBE}_1$ has a large security loss and requires a strong LWE assumption. Since both the improvement and the downside are inherited from the concrete Type-II PHF construction, this situation can be immediately changed if one can find a better lattice-based PHF.

1.5 Other Related Work

Hofheinz and Kiltz [35] first introduced the notion of PHF as a special keyed 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. [55] 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 $(\mathrm{poly}, 1)$-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

[5] Informally, an algorithm is algebraic if there is way to compute the representation of a group element output by the algorithm in terms of its input group elements [13].

called asymmetric PHF over bilinear maps, and used it to construct (homomorphic) signature schemes with short verification keys.

All the above PHF constructions [35,34,55,28,17] seem specific to groups with nice properties, which might constitute a main barrier to instantiate them from lattices. Although several lattice-based schemes [1,16] had employed a similar partitioning proof trick as that was captured by the traditional PHFs, it was still 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 [1,14,45].

Very recently, two current and independent papers [54,7] also focused on designing fully secure IBE with short master public keys. Specifically, Yamada [54] constructed an efficient IBE scheme with master public key roughly consisting of $O(\ell^{1/d})$ matrices for some constant $d > 0$, where ℓ is the identity length (or the output length of the hash function for identities) and is set to be equal to the security paramter n, i.e., $\ell = n$. In [7], Apon, Fan and Liu proposed an IBE scheme as compact as PKE by using a 'restricted' setting of $\ell = \omega(\log n)$. For comparison, the master public key of our IBE scheme $\mathcal{IBE}_1$ consists of $O(\log n)$ matrices for the choice of $\ell = n$. Besides, both IBE constructions in [54,7] seem to implicitly use some kind of lattice-based PHFs.

1.6 Roadmap

After some preliminaries in Section 2, we give the definition of lattice-based PHFs, and two types of constructions in Section 3. We construct signatures and IBEs from lattice-based PHFs in Section 4 and Section 5, respectively. A short conclusion is given in Section 6.

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 $\mathrm{poly}(n)$ 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 $\leftarrow_r$ 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 $0 \leq v \leq q$, we denote $\mathsf{BitDecomp}_q(v) \in \{0,1\}^k$ as the k-dimensional bit-decomposition of v, where $k = \lceil \log_2 q \rceil$. 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 [1,22], FRD encodings supporting the exponential size domain $\mathbb{Z}_q^n$ can be efficiently constructed.

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 [46,31,48].

Lemma 1. *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*

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

Following [45,26], 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 $\mathbb{E}(\exp(2\pi tX))\leq\exp(\pi s^2t^2)$. 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 [51].

Lemma 2. *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 $2\exp(-\pi t^2)$.*

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: $\mathbf{\Lambda}_q^\perp(\mathbf{A})=\{\mathbf{e}\in\mathbb{Z}^m \ s.t.\ \mathbf{A}\mathbf{e}=\mathbf{0} \mod q\}$ and $\mathbf{\Lambda}_q(\mathbf{A})=\{\mathbf{y}\in\mathbb{Z}^m \ s.t.\ \exists\mathbf{s}\in\mathbb{Z}^n,\ \mathbf{A}^\mathbf{t}\mathbf{s}=\mathbf{y} \mod q\}$. By definition, we have $\mathbf{\Lambda}_q^\perp(\mathbf{A})=\mathbf{\Lambda}_q^\perp(\mathbf{C}\mathbf{A})$ for any invertible $\mathbf{C}\in\mathbb{Z}_q^{n\times n}$. 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 [45]. Let $\mathbf{I}_n$ be the $n\times n$ identity matrix. We now recall the publicly known trapdoor matrix $\mathbf{G}$ in [45]. Formally, for any prime $q>2$, integer $n\geq 1$ and $k=\lceil\log_2 q\rceil$, define $\mathbf{g}=(1,2,\ldots,2^{k-1})^t\in\mathbb{Z}_q^k$ and $\mathbf{G}=\mathbf{I}_n\otimes\mathbf{g}^t\in\mathbb{Z}_q^{n\times nk}$, where '$\otimes$' represents the tensor product.[6] Then, the lattice $\mathbf{\Lambda}_q^\perp(\mathbf{G})$ has a publicly known short basis $\mathbf{T}=\mathbf{I}_n\otimes\mathbf{T}_k\in\mathbb{Z}^{nk\times nk}$ with $\|\mathbf{T}\|\leq\max\{\sqrt{5},\sqrt{k}\}$. Let $(q_0,q_1,\ldots,q_{k-1})=\mathsf{BitDecomp}_q(q)\in\{0,1\}^k$, we have

[6] One can define $\mathbf{G}$ by using any base $b\geq 2$ and $\mathbf{g}=(1,b,\ldots,b^{k-1})^t$ for $k=\lceil\log_b q\rceil$. In this paper, we fix $b=2$ for simplicity.

$$\mathbf{G} = \begin{pmatrix} \cdots \mathbf{g}^t \cdots & & & \\ & \cdots \mathbf{g}^t \cdots & & \\ & & \ddots & \\ & & & \cdots \mathbf{g}^t \cdots \end{pmatrix} \qquad \mathbf{T}_k = \begin{pmatrix} 2 & & & & q_0 \\ -1 & 2 & & & q_1 \\ & -1 & & & q_2 \\ & & \ddots & & \vdots \\ & & & 2 & q_{k-2} \\ & & & -1 & q_{k-1} \end{pmatrix}$$

For any vector $\mathbf{u} \in \mathbb{Z}_q^n$, the basis $\mathbf{T} = \mathbf{I}_n \otimes \mathbf{T}_k \in \mathbb{Z}_q^{nk \times nk}$ can be used to sample short vector $\mathbf{e} \sim D_{\mathbb{Z}^{nk},s}$ satisfying $\mathbf{Ge} = \mathbf{u}$ for any $s \geq \omega(\sqrt{\log n})$ in quasilinear time. Besides, one can deterministically compute a short vector $\mathbf{v} = \mathbf{G}^{-1}(\mathbf{u}) \in \{0,1\}^{nk}$ such that $\mathbf{Gv} = \mathbf{u}$. This fact will be frequently used in this paper.

Definition 1 (G-trapdoor [45]). *For any integers $n, \bar{m}, q \in \mathbb{Z}, k = \lceil \log_2 q \rceil$, 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 $\mathbf{A}\begin{bmatrix} \mathbf{R} \\ \mathbf{I}_{nk} \end{bmatrix} = \mathbf{SG}$ 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 [45].

Proposition 1 ([45]). *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 $\mathsf{TrapGen}(1^n, 1^{\bar{m}}, q, \mathbf{S})$ 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 $\mathbf{A}\begin{bmatrix} \mathbf{R} \\ \mathbf{I}_{nk} \end{bmatrix} = \mathbf{SG}$, where $k = \lceil \log_2 q \rceil$.*

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 $\mathsf{SampleD}(\mathbf{R}, \mathbf{A}, \mathbf{S}, \mathbf{U}, s)$ that samples from a distribution within $\mathrm{negl}(n)$ statistical distance of $\mathbf{E} \sim (D_{\mathbb{Z}^{\bar{m}},s})^{n'}$ satisfying $\mathbf{AE} = \mathbf{U}$.

We also need the following useful facts from [48,31,45].

Lemma 3. *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 $\mathbf{A} \leftarrow_r \mathbb{Z}_q^{n \times m}$, 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 $\mathbf{y} \in \mathbf{\Lambda}_q^{\perp}(\mathbf{A})$, $\Pr_{\mathbf{x} \leftarrow_r D_{\mathbf{\Lambda}_q^{\perp}(\mathbf{A}),s,\mathbf{c}}}[\mathbf{x} = \mathbf{y}] \leq 2^{1-m}$;*
- *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 $\mathbf{v} + D_{\mathbf{\Lambda}_q^{\perp}(\mathbf{A}),s,-\mathbf{v}}$.*

2.3 Learning with Errors (LWE) and Small Integer Solutions (SIS)

For any positive integer n, q, real $\alpha > 0$, and any vector $\mathbf{s} \in \mathbb{Z}_q^n$, the distribution $A_{\mathbf{s},\alpha}$ over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ is defined as $A_{\mathbf{s},\alpha} = \{(\mathbf{a}, \mathbf{a}^t\mathbf{s} + x \mod q) : \mathbf{a} \leftarrow_r \mathbb{Z}_q^n, x \leftarrow_r D_{\mathbb{Z},\alpha q}\}$, where $D_{\mathbb{Z},\alpha q}$ is the discrete Gaussian distribution over $\mathbb{Z}$ 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 $(\mathbf{A}, \mathbf{y}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$, 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 if, for uniformly random $\mathbf{s} \leftarrow_r \mathbb{Z}_q^n$, 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 $\mathbf{s} \leftarrow_r \mathbb{Z}_q^n$, the solving algorithm is asked to distinguish $A_{\mathbf{s},\alpha}$ from the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ (with only polynomial samples). For certain modulus q, the average-case decisional LWE problem is polynomially equivalent to its worst-case search version [49].

Proposition 2 ([49]). *Let $\alpha = \alpha(n) \in (0,1)$ and let $q = q(n)$ be a prime such that $\alpha q > 2\sqrt{n}$. If there exists an efficient (possibly quantum) algorithm that solves* $\mathrm{LWE}_{q,\alpha}$*, then there exists an efficient quantum algorithm for approximating* SIVP *(in the l_2 norm) on n-dimensional lattices, in the worst case, to within $\tilde{O}(n/\alpha)$ factors.*

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 $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, the $\mathrm{SIS}_{q,m,\beta}$ problem asks to find a non-zero vector $\mathbf{e} \in \mathbb{Z}^m$ such that $\mathbf{Ae} = \mathbf{0} \mod q$ and $\|\mathbf{e}\| \le \beta$. In [31], Gentry et al. introduced the ISIS problem, which was an inhomogeneous variant of SIS. Specifically, given an extra random syndrome $\mathbf{u} \in \mathbb{Z}_q^n$, the $\mathrm{ISIS}_{q,m,\beta}$ problem asks to find a vector $\mathbf{e} \in \mathbb{Z}^m$ such that $\mathbf{Ae} = \mathbf{u} \mod q$ and $\|\mathbf{e}\| \le \beta$. Both the two problems were shown to be as hard as certain worst-case lattice problems [31].

Proposition 3 ([31]). *For any polynomially bounded $m, \beta = poly(n)$ and prime $q \ge \beta \cdot \omega(\sqrt{n \log n})$, the average-case problems* $\mathrm{SIS}_{q,m,\beta}$ *and* $\mathrm{ISIS}_{q,m,\beta}$ *are as hard as approximating* SIVP *on n-dimensional lattices, in the worst case, to within certain $\gamma = \beta \cdot \widetilde{O}(\sqrt{n})$ factors.*

3 Programmable Hash Functions from Lattices

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

Definition 2 (Lattice-based Programmable Hash Function). *A hash function $\mathcal{H} : \mathcal{X} \to \mathbb{Z}_q^{n\times m}$ is a $(u, v, \beta, \gamma, \delta)$-*PHF *if there exist a PPT trapdoor key generation algorithm $\mathcal{H}.$*TrapGen *and an efficiently deterministic trapdoor evaluation algorithm $\mathcal{H}.$*TrapEval *such that given a uniformly random $\mathbf{A} \in \mathbb{Z}_q^{n\times\bar{m}}$ and a (public) trapdoor matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$,*[7] *the following properties hold:*

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

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

[7] A general trapdoor matrix $\mathbf{B}$ is used for utmost generality, but the publicly known trapdoor matrix $\mathbf{B} = \mathbf{G}$ in [45] is recommended for both efficiency and simplicity.

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

Well-distributed hidden matrices: *For all* $(K', td) \leftarrow \mathcal{H}.\mathrm{TrapGen}(1^\kappa, \mathbf{A}, \mathbf{B})$*, 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* $(\mathbf{R}_{Y_i}, \mathbf{S}_{Y_i}) = \mathcal{H}.\mathrm{TrapEval}(td, K', Y_i)$*. Then, we have that*

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

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* u *(resp.* v*) is an arbitrary polynomial in* κ*, we say that* $\mathcal{H}$ *is a* $(\mathrm{poly}, v, \beta)$-PHF *(resp.* $(u, \mathrm{poly}, \beta)$-PHF*).*

A ***weak programmable hash function*** is a relaxed version of PHF, where the $\mathcal{H}.\mathrm{TrapGen}$ 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 $\mathcal{H} : \mathcal{X} \to \mathbb{Z}_q^{n \times m}$ 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 3. *Let* $\ell, n, m, q \in \mathbb{Z}$ *be some polynomials in the security parameter* κ*. Let* E *be a deterministic encoding from* $\mathcal{X}$ *to* $(\mathbb{Z}_q^{n \times n})^\ell$*, the hash function* $\mathcal{H} = (\mathcal{H}.\mathrm{Gen}, \mathcal{H}.\mathrm{Eval})$ *with key space* $\mathcal{K} \subseteq (\mathbb{Z}_q^{n \times m})^{\ell+1}$ *is defined as follows:*

- $\mathcal{H}.\mathrm{Gen}(1^\kappa)$*: Randomly choose* $(\mathbf{A}_0, \ldots, \mathbf{A}_\ell) \leftarrow_r \mathcal{K}$*, 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, [14,10,47]) and encryptions (e.g., [1,45]). Let $\mathbf{I}_n$ be the $n \times n$ identity matrix. In the following theorems, we summarize several known results which were implicitly proved in [14,1,45].

Theorem 1. *Let* $\mathcal{K} = (\mathbb{Z}_q^{n \times m})^{\ell+1}$ *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* $\mathbf{C}_i = (-1)^{X_i} \cdot \mathbf{I}_n$ *for* $i = \{1, \ldots, \ell\}$*. Then, for large enough integer* $\bar{m} = O(n \log q)$ *and any fixed polynomial* $v = v(\kappa) \in \mathbb{Z}$*, the instantiated hash function* $\mathcal{H}$ *of Definition 3 is a* $(1, v, \beta, \gamma, \delta)$-PHF *with* $\beta \leq \sqrt{\ell \bar{m}} \cdot \omega(\sqrt{\log n})$*,* $\gamma = \mathrm{negl}(\kappa)$ *and* $\delta = \frac{1}{q^t}(1 - \frac{v}{q^t})$*, where* t *is the smallest integer satisfying* $q^t > 2v$*.*

Theorem 2. *For large enough* $\bar{m} = O(n \log q)$*, the hash function* $\mathcal{H}$ *given in Definition 3 is a weak* $(1, \mathrm{poly}, \beta, \gamma, \delta)$-PHF *with* $\beta \leq \sqrt{\ell \bar{m}} \cdot \omega(\sqrt{\log n})$*,* $\gamma = \mathrm{negl}(\kappa)$*, and* $\delta = 1$ *when instantiated as follows:*

- *Let* $\mathcal{K} = (\mathbb{Z}_q^{n \times m})^2$ *(i.e.,* $\ell = 1$*) and* $\mathcal{X} = \mathbb{Z}_q^n$*. Given an input* $X \in \mathcal{X}$*, the encoding* $\mathrm{E}(X)$ *returns* $H(X)$ *where* $H : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n \times n}$ *is an FRD encoding.*

– *Let* $\mathcal{K} = (\mathbb{Z}_q^{n\times m})^{\ell+1}$ *and* $\mathcal{X} = \{0,1\}^\ell$*. Given an input* $X = (X_1, \ldots, X_\ell) \in \mathcal{X}$*, the encoding* $\mathrm{E}(X)$ *returns* $\mathbf{C}_i = X_i \cdot \mathbf{I}_n$ *for all* $i \in \{1, \ldots, \ell\}$*.*

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) $(1, v, \beta)$-PHFs for some polynomial $v \in \mathbb{Z}$ and real $\beta \in \mathbb{R}$.

3.2 Type-II Construction

Let integers $\ell, \bar{m}, n, q, u, v, L, N$ 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 $\mathbf{B} = \mathbf{G} \in \mathbb{Z}_q^{n\times nk}$ to construct more efficient PHF from lattices for any $v = \mathrm{poly}(\kappa)$. We begin by first recalling the notion of cover-free sets. Formally, 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 [27,40]. Formally,

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

In the following, we use the binary representation of $[N]$ to construct lattice-based PHFs with short keys.

Definition 4. *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 4. Let* $\mu = \lceil \log_2 N \rceil$ *and* $k = \lceil \log_2 q \rceil$*. Then, the hash function* $\mathcal{H} = (\mathcal{H}.\mathrm{Gen}, \mathcal{H}.\mathrm{Eval})$ *from* $[L]$ *to* $\mathbb{Z}_q^{n\times nk}$ *is defined as follows:*

– $\mathcal{H}.\mathrm{Gen}(1^\kappa)$*: Randomly choose* $\hat{\mathbf{A}}, \mathbf{A}_i \leftarrow_r \mathbb{Z}_q^{n\times nk}$ *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* $K = (\hat{\mathbf{A}}, \{\mathbf{A}_i\}_{i\in\{0,\ldots,\mu-1\}})$ *and integer* $X \in [L]$*, the algorithm performs the* **Procedure I** *in* Fig. 1 *to compute* $\mathbf{Z} = \mathrm{H}_K(X)$*.*

We now show that for any prior fixed $v = \mathrm{poly}(\kappa)$, the hash function $\mathcal{H}$ given in Definition 4 is a $(1, v, \beta)$-PHF for some polynomially bounded $\beta \in \mathbb{R}$.

Theorem 3. *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 4. Then, for large enough* $\bar{m} = O(n \log q)$*, the hash function* $\mathcal{H}$ *in Definition 4 is a* $(1, v, \beta, \gamma, \delta)$-PHF *with* $\beta \leq \mu v\ell\bar{m}^{1.5} \cdot \omega(\sqrt{\log \bar{m}})$*,* $\gamma = \mathrm{negl}(\kappa)$ *and* $\delta = 1/N$*, where* $\mu = \lceil \log_2 N \rceil$*.*

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* $\mu = O(\log n)$ *matrices.*

Procedure I

$\mathbf{Z} := \hat{\mathbf{A}}$
For all $z \in CF_X$
  $(b_0, \ldots, b_{\mu-1}) := \mathsf{BitDecomp}_N(z)$
  $\mathbf{B}_z := \mathbf{A}_{\mu-1} - b_{\mu-1} \cdot \mathbf{G}$
  For $i = \mu - 2, \ldots, 0$
    $\mathbf{B}_z := (\mathbf{A}_i - b_i \cdot \mathbf{G}) \cdot \mathbf{G}^{-1}(\mathbf{B}_z)$
  $\mathbf{Z} := \mathbf{Z} + \mathbf{B}_z$
Return $\mathbf{Z}$

Procedure II

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

Fig. 1. The Procedures Used in Definition 4 and Theorem 3

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

- $\mathcal{H}$.TrapGen$(1^\kappa, \mathbf{A}, \mathbf{G})$: Given a uniformly random $\mathbf{A} \in \mathbb{Z}_q^{n \times \bar{m}}$ and matrix $\mathbf{G} \in \mathbb{Z}_q^{n \times nk}$ for sufficiently large $\bar{m} = O(n \log q)$, let $s \geq \omega(\sqrt{\log \bar{m}}) \in \mathbb{R}$ satisfy the requirement in Lemma 3. Randomly choose $\hat{\mathbf{R}}, \mathbf{R}_i \leftarrow_r (D_{\mathbb{Z}^{\bar{m}}, s})^{nk}$ for $i \in \{0, \ldots, \mu-1\}$, and an integer $z^* \leftarrow_r [N]$. 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 $td = (\hat{\mathbf{R}}, \{\mathbf{R}_i\}_{i \in \{0, \ldots, \mu-1\}}, z^*)$.
- $\mathcal{H}$.TrapEval(td, K', X): Given td and an input $X \in [L]$, the algorithm first computes CF_X by Lemma 4. Then, let $(b^*_0, \ldots, b^*_{\mu-1}) = \mathsf{BitDecomp}_N(z^*)$, and perform the **Procedure II** in Fig. 1 to compute $(\mathbf{R}_X, \mathbf{S}_X)$.

Since $s \geq \omega(\sqrt{\log \bar{m}})$ and $\hat{\mathbf{R}}, \mathbf{R}_i \leftarrow_r (D_{\mathbb{Z}^{\bar{m}}, s})^{nk}$, each matrix in key $K' = (\hat{\mathbf{A}}, \{\mathbf{A}_i\}_{i \in \{0, \ldots, \mu-1\}})$ is statistically close to uniform over $\mathbb{Z}_q^{n \times nk}$ by Lemma 3. 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. 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 inner 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 inner 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 $\mathbf{S}_z = \prod_{i=0}^{\mu-1} (1 - b^*_i - b_i) \cdot \mathbf{I}_n$ 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 $\mathbf{S}_z = (-1)^c \cdot \mathbf{I}_n$ for $z = z^*$, where c is the number of 1's in the binary vector $(b^*_0, \ldots, b^*_{\mu-1}) = \mathsf{BitDecomp}_N(z^*)$. 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 $\mathbf{S}_X = -(-1)^c \cdot \mathbf{I}_n$ if $z^* \notin CF_X$, else $\mathbf{S}_X = \mathbf{0}$.

Since $s_1(\mathbf{G}^{-1}(\mathbf{B}_z)) \leq nk$ by the fact that $\mathbf{G}^{-1}(\mathbf{B}_z) \in \{0,1\}^{nk \times nk}$, and $s_1(\hat{\mathbf{R}}), s_1(\mathbf{R}_i) \leq (\sqrt{\bar{m}} + \sqrt{nk}) \cdot \omega(\sqrt{\log \bar{m}})$ by Lemma 2, we have that $s_1(\mathbf{R}_z) \leq \mu \bar{m}^{1.5} \cdot \omega(\sqrt{\log \bar{m}})$ holds except

with negligible probability for any $z \in CF_X$. Using $|CF_X| = \eta \leq 4v\ell$, the inequality $s_1(\mathbf{R}_X) \leq \mu v\ell\bar{m}^{1.5} \cdot \omega(\sqrt{\log \bar{m}})$ holds except with negligible probability for any $X \in [L]$. Besides, for any $X_1, Y_1, \ldots, Y_v \in [L]$ 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 $\cup_{j\in\{1,\ldots,v\}} CF_{Y_j}$. 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 $\Pr[z^* \in CF_{X_1} \wedge z^* \notin \cup_{j\in\{1,\ldots,v\}} CF_{Y_j}]$ is at least $1/N$. Thus, we have that $\Pr[\mathbf{S}_{X_1} = \mathbf{0} \wedge \mathbf{S}_{Y_1} = \cdots = \mathbf{S}_{Y_v} = -(-1)^c \cdot \mathbf{I}_n \in \mathcal{I}_n] \geq \frac{1}{N}$. □

3.3 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

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

is negligible in the security parameter κ.

Theorem 4. *Let $n, v, q \in \mathbb{Z}$ and $\bar{\beta}, \beta \in \mathbb{R}$ be polynomials in the security parameter κ. Let $\mathcal{H} = (\mathcal{H}.\mathrm{Gen}, \mathcal{H}.\mathrm{Eval})$ be a $(1, v, \beta, \gamma, \delta)$-PHF 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 for $\bar{\beta} = \beta\sqrt{m} \cdot \omega(\log n)$ with probability at least $\epsilon' \geq (\epsilon - \gamma)\delta$.*

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 $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$ be any trapdoor matrix that allows to efficiently sample short vector $\mathbf{v} \in \mathbb{Z}^m$ such that $\|\mathbf{v}\| \leq \sqrt{m} \cdot \omega(\log n)$ and $\mathbf{Bv} = \mathbf{u}'$ for any $\mathbf{u}' \in \mathbb{Z}_q^n$ (e.g., $\mathbf{B}$ is generated by using the trapdoor generation algorithm in Proposition 1). Formally, 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 $\mathcal{B}$ computes $(K', td) \leftarrow_r \mathcal{H}.\mathrm{TrapGen}(1^\kappa, \mathbf{A}, \mathbf{B})$, 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 $\mathrm{H}_{K'}(X_1) = \mathbf{AR}_{X_1} + \mathbf{S}_{X_1}\mathbf{B} = \mathbf{AR}_{X_2} + \mathbf{S}_{X_2}\mathbf{B} = \mathrm{H}_{K'}(X_2)$. 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 $\mathbf{AR}_{X_1} = \mathbf{AR}_{X_2} + \mathbf{S}_{X_2}\mathbf{B}$ 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 $\mathbf{v} \in \mathbb{Z}_q^m$ is sampled by using the trapdoor of $\mathbf{B}$ such that $\|\mathbf{v}\| \leq \sqrt{m} \cdot \omega(\log n)$ and $\mathbf{Bv} = \mathbf{S}_{X_2}^{-1}\mathbf{u}$. By $\mathbf{Ax} = \mathbf{S}_{X_2}\mathbf{Bv} = \mathbf{u}$, we have that $\mathbf{x}$ is a solution of $\mathbf{Ax} = \mathbf{u}$. In addition, since $s_1(\mathbf{R}_{X_1}), s_1(\mathbf{R}_{X_2}) \leq \beta$ by assumption, we have $\|\mathbf{x}\| \leq \beta\sqrt{m} \cdot \omega(\log n)$. This completes the proof. □

High Min-Entropy. Let $\mathcal{H} : \mathcal{X} \to \mathbb{Z}_q^{n\times m}$ 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 $\mathrm{H}_{K'}(X) = \mathbf{AR}_X + \mathbf{S}_X\mathbf{B}$ 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 5 (PHF with High Min-Entropy). *Let $\mathcal{H} : \mathcal{X} \to \mathbb{Z}_q^{n\times m}$ 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 $\mathcal{H}$.*TrapGen *and $\mathcal{H}$.*TrapEval *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 $\mathbf{A} \in \mathbb{Z}_q^{n\times \bar{m}}$ and (publicly known) trapdoor matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$, the following conditions hold*

1. *For any $(K', td) \leftarrow \mathcal{H}.\mathrm{TrapGen}(1^\kappa, \mathbf{A}, \mathbf{B}), K \leftarrow \mathcal{H}.\mathrm{Gen}(1^\kappa)$, any $X \in \mathcal{X}$ and any $\mathbf{w} \in \mathbb{Z}_q^{\bar{m}}$, 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 $(\mathbf{R}_X, \mathbf{S}_X) = \mathcal{H}.\mathrm{TrapEval}(td, K', X)$.*
2. *For any $(K', td) \leftarrow \mathcal{H}.\mathrm{TrapGen}(1^\kappa, \mathbf{A}, \mathbf{B})$, any $X \in \mathcal{X}$, any uniformly random $\mathbf{v} \in \mathbb{Z}_q^{\bar{m}}$, and any uniformly random $\mathbf{u} \leftarrow_r \mathbb{Z}_q^m$, 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 $(\mathbf{R}_X, \mathbf{S}_X) = \mathcal{H}.\mathrm{TrapEval}(td, K', X)$.*

Remark 1. 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 [1] implicitly showed that the Type-I PHF construction satisfied the high min-entropy property. Now, we show that our Type-II PHF construction also has the high min-entropy property.

Theorem 5. *Let integers $n, \bar{m}, q$ 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 $N \le 16v^2\ell, \eta \le 4v\ell$ and $CF = \{CF_X\}_{X\in[L]}$ be defined as in Lemma 4. Then, for large enough $\bar{m} = O(n \log q)$, the hash function $\mathcal{H} : [L] \to \mathbb{Z}_q^{n\times nk}$ given in Definition 4 (and proved in Theorem 3) is a PHF with high min-entropy.*

Proof. By Definition 4, the real key K of $\mathcal{H}$ is uniformly distributed over $(\mathbb{Z}_q^{n\times nk})^{2\mu+1}$. To prove that $\mathcal{H}$ satisfies the first condition of high min-entropy, we must show that for any $(K', td) \leftarrow \mathcal{H}.\mathrm{TrapGen}(1^\kappa, \mathbf{A}, \mathbf{G})$, any $X \in \mathcal{X}$ and $(\mathbf{R}_X, \mathbf{S}_X) = \mathcal{H}.\mathrm{TrapEval}(td, K', X)$, the key K' is statistically close to uniform over $(\mathbb{Z}_q^{n\times nk})^{2\mu+1}$ even conditioned on $\mathbf{R}_X^t\mathbf{w} \in \mathbb{Z}_q^{nk}$. Formally, for any $\mathbf{w} \in \mathbb{Z}_q^{\bar{m}}$, let $f_\mathbf{w} : \mathbb{Z}_q^{\bar{m}\times nk} \to \mathbb{Z}_q^{nk}$ be the function defined by $f_\mathbf{w}(\mathbf{X}) = \mathbf{X}^t\mathbf{w} \in \mathbb{Z}_q^{nk}$. Then, given $I = \{f_\mathbf{w}(\hat{\mathbf{R}}), \{f_\mathbf{w}(\mathbf{R}_i)\}_{i\in\{0,\ldots,\mu-1\}}\}$ and (K', X, z^*), one can compute $\mathbf{R}_X^t\mathbf{w}$ by simulating the **Procedure II** in Theorem 3. Thus, it suffices to show that K' is statistically close to uniform over $(\mathbb{Z}_q^{n\times nk})^{2\mu+1}$ conditioned on I and z^*. Since each matrix in the key K' always has a form of $\mathbf{A}\tilde{\mathbf{R}} + b\mathbf{G}$ for some randomly chosen $\tilde{\mathbf{R}} \leftarrow_r (D_{\mathbb{Z}^{\bar{m}},s})^{nk}$, and a bit $b \in \{0, 1\}$ depending on a random $z^* \leftarrow_r [N]$. Using a standard hybrid argument, it is enough to show that conditioned on $\mathbf{A}$ and $f_\mathbf{w}(\hat{\mathbf{R}})$, $\mathbf{A}\tilde{\mathbf{R}}$ is statistically close to uniform over $\mathbb{Z}_q^{n\times nk}$.

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

Now, we show that $\mathcal{H}$ satisfies the second condition in Definition 5. 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 $\hat{\mathbf{R}} \leftarrow_r (D_{\mathbb{Z}^{\bar{m}},s})^{nk}$ 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 $\bar{m} = O(n \log q)$, the pair $(\mathbf{A}\hat{\mathbf{R}}, \hat{\mathbf{u}}^t = \mathbf{v}^t\hat{\mathbf{R}})$ is statistically close to uniform over $\mathbb{Z}_q^{n \times nk} \times \mathbb{Z}_q^{nk}$ by the fact in Lemma 3.[8] Thus, $\mathbf{R}_X^t\mathbf{v} = \hat{\mathbf{R}}^t\mathbf{v} + \tilde{\mathbf{R}}^t\mathbf{v}$ is uniformly distributed over $\mathbb{Z}_q^{nk}$. This completes the proof of Theorem 5. □

3.4 Programmable Hash Function from Ideal Lattices

As many cryptographic schemes over general lattices (e.g., [45]), we do not see any obstacle preventing us from adapting our definition and constructions of PHFs to ideal lattices defined over polynomial rings, e.g., $R = \mathbb{Z}[x]/(x^n + 1)$ or $R_q = \mathbb{Z}_q[x]/(x^n + 1)$ where n is a power of 2. In general, one can benefit from the rich algebraic structures of ideal lattices in many aspects. For example, compared to their counterparts over general lattices, the constructions over ideal lattices roughly save a factor of n in the key size (e.g., [43,44]).

At CRYPTO 2014, Ducas and Micciancio [26] proposed a short signature scheme by combining the confined guessing technique [10] with ideal lattices, which substantially reduced the verification key size from previous known $O(n)$ elements to $O(\log n)$ elements. We note that their construction implicitly used a weak $(1, \text{poly}, \beta)$-PHF for some $\beta = \text{poly}(\kappa) \in \mathbb{R}$ (we omit the details for not involving too many backgrounds on ideal lattices). But as noted by the authors, their methods used for constructing signatures with short verification keys (as well as the underlying PHF) seem specific to the ideal lattice setting, and thus cannot be instantiated from general lattices. In fact, it was left as an open problem [26] to construct a standard model short signature scheme with short verification keys from general lattices.

4 Short Signature Schemes from Lattice-based PHFs

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 fresh 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 times.

[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 3.

Forge. $\mathcal{F}$ outputs a message-signature pair (M^*, σ^*). Let Q be the set of all messages queried by $\mathcal{F}$ in the signing phase. If $M^* \notin Q$ 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 $\mathrm{Adv}^{\text{euf-cma}}_{\mathcal{SIG},\mathcal{F}}(1^\kappa) = \Pr[\mathcal{C} \text{ outputs } 1]$.

Definition 6 (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 $\mathrm{Adv}^{\text{euf-cma}}_{\mathcal{SIG},\mathcal{F}}(1^\kappa)$ 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 $\{M_1, \ldots, M_Q\}$ 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 6 is denoted as EUF-naCMA. One can transform an EUF-naCMA secure signature scheme into an EUF-CMA secure one [10,26] by using chameleon hash functions [39].

4.1 A Short Signature Scheme with Short Verification Key

Let integers $\ell, n, m', v, q \in \mathbb{Z}, \beta \in \mathbb{R}$ be some polynomials in the security parameter κ, and let $k = \lceil \log_2 q \rceil$. Let $\mathcal{H} = (\mathcal{H}.\mathrm{Gen}, \mathcal{H}.\mathrm{Eval})$ be any $(1, v, \beta)$-PHF from $\{0,1\}^\ell$ to $\mathbb{Z}_q^{n \times m'}$. Let $\bar{m} = O(n \log q)$, $m = \bar{m} + m'$, 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, compute $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^n, 1^{\bar{m}}, q, \mathbf{I}_n)$ such that $\mathbf{A} \in \mathbb{Z}_q^{n \times \bar{m}}$, $\mathbf{R} = \mathbb{Z}_q^{(\bar{m}-nk) \times nk}$, and randomly choose $\mathbf{u} \leftarrow_r \mathbb{Z}_q^n$. Then, compute $K \leftarrow \mathcal{H}.\mathrm{Gen}(1^\kappa)$, and return a pair of verification key and secret signing key $(vk, sk) = ((\mathbf{A}, \mathbf{u}, K), \mathbf{R})$.

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

$\mathsf{Verify}(vk, M, \sigma)$**:** Given vk, a message M and a vector $\sigma = \mathbf{e}$, compute $\mathbf{A}_M = (\mathbf{A} \| \mathrm{H}_K(M)) \in \mathbb{Z}_q^{n \times m}$, where $\mathrm{H}_K(M) = \mathcal{H}.\mathrm{Eval}(K, M) \in \mathbb{Z}_q^{n \times m'}$. Return 1 if $\|\mathbf{e}\| \leq s\sqrt{m}$ 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 [14,45] 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 PHF construction in Definition 4 to obtain a signature scheme with verification keys consisting of a logarithmic number of matrices. As for the security, we have the following theorem.

Theorem 6. *Let $\ell, n, \bar{m}, m', q \in \mathbb{Z}$ and $\bar{\beta}, \beta, s \in \mathbb{R}$ be some polynomials in the security parameter κ, and let $m = \bar{m} + m'$. Let $\mathcal{H} = (\mathcal{H}.\mathrm{Gen}, \mathcal{H}.\mathrm{Eval})$ be a $(1, v, \beta, \gamma, \delta)$-PHF from $\{0,1\}^\ell$ to $\mathbb{Z}_q^{n \times m'}$ with $\gamma = \mathrm{negl}(\kappa)$ and noticeable $\delta > 0$. Then, for large enough $\bar{m} = O(n \log q)$ and $s > \max(\beta, \sqrt{m}) \cdot \omega(\sqrt{\log n}) \in \mathbb{R}$, if there exists a PPT forger $\mathcal{F}$ breaking the* EUF-CMA *security of $\mathcal{SIG}$ with non-negligible probability $\epsilon > 0$ and making at most $Q \leq v$ signing queries, there exists an algorithm $\mathcal{B}$ solving the $\mathrm{ISIS}_{q,\bar{m},\bar{\beta}}$ problem for $\bar{\beta} = \beta s \sqrt{m} \cdot \omega(\sqrt{\log n})$ with probability at least $\epsilon' \geq \epsilon\delta - \mathrm{negl}(\kappa)$.*

[9] Note that the scheme in [14] used a syndrome $\mathbf{u} = \mathbf{0}$, we prefer to use a random chosen syndrome $\mathbf{u} \leftarrow_r \mathbb{Z}_q^n$ as that in [45] for simplifying the security analysis.

Since a proof sketch is given in Section 1.3, we omit the details of the proof. Let $\mathcal{SIG}_1$ denote the signature scheme obtained by instantiating $\mathcal{SIG}$ with our Type-II PHF construction in Definition 4. Then, the verification key of $\mathcal{SIG}_1$ has $O(\log n)$ matrices 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 $\bar{m} = O(n \log q), v = \mathrm{poly}(n)$ and $\ell = n$. If there exists a PPT forger $\mathcal{F}$ breaking the* EUF-CMA *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 for $\bar{\beta} = v^2 \cdot \tilde{O}(n^{5.5})$ with probability at least $\epsilon' \geq \frac{\epsilon}{16nv^2} - \mathrm{negl}(\kappa)$.*

4.2 An Improved Short Signature Scheme from Weaker Assumption

Compared to prior constructions in [10,26,6], our $\mathcal{SIG}_1$ only has a reduction loss about $16nQ^2$, which does not depend on the forger's success probability ϵ. However, because of $v \geq Q$, our improvement requires the $\mathrm{ISIS}_{q,\bar{m},\bar{\beta}}$ problem to be hard for $\bar{\beta} = Q^2 \cdot \tilde{O}(n^{5.5})$, which means that the modulus q should be bigger than $Q^2 \cdot \tilde{O}(n^{5.5})$. Even though q is still a polynomial of n in an asymptotic sense, it might be very large in practice. In this section, we further remove 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 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 as long as the probability that $\mathbf{S}'_{\mathbf{t}} + \mathbf{S}_M \in \mathcal{I}_n$ is invertible for all the Q signing messages, but $\mathbf{S}'_{\mathbf{t}^*} + \mathbf{S}_{M^*} = \mathbf{0}$ 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 $\mathcal{H}'$ in Theorem 2 and the Type-II PHF $\mathcal{H} = (\mathcal{H}.\mathrm{Gen}, \mathcal{H}.\mathrm{Eval})$ given in Definition 4. 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 [1] 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 $\mathcal{SIG}_2 = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ is defined as follows.

$\mathsf{KeyGen}(1^\kappa)$**:** Given a security parameter κ, 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 $\mathbf{A}_0 \leftarrow_r \mathbb{Z}_q^{n \times nk}$ and $\mathbf{u} \leftarrow_r \mathbb{Z}_q^n$. Finally, compute $K \leftarrow \mathcal{H}.\mathrm{Gen}(1^\kappa)$, and return $(vk, sk) = ((\mathbf{A}, \mathbf{A}_0, \mathbf{u}, K), \mathbf{R})$.

$\mathsf{Sign}(sk, M \in \{0,1\}^n)$**:** Given the secret key sk and a message M, randomly choose $\mathbf{t} \leftarrow_r \{0,1\}^\ell$, 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 $\sigma = (\mathbf{e}, \mathbf{t})$, 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 1. This shows that $\mathcal{SIG}_2$ is correct.

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

Theorem 7. *Let $\ell, \bar{m}, n, q, v \in \mathbb{Z}$ be polynomials in the security parameter κ. For appropriate choices of $\ell = O(\log n)$ and $v = \omega(\log n)$, if there exists a PPT forger $\mathcal{F}$ breaking the* EUF-CMA *security of $\mathcal{SIG}_2$ with non-negligible probability ϵ and making at most $Q = \mathrm{poly}(n)$ 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 $\frac{\epsilon}{\tilde{O}(n^2)}$. Formally, $\mathcal{B}$ first randomly chooses a vector $\mathbf{t}' \leftarrow_r \{0,1\}^\ell$ and hopes that $\mathcal{F}$ will output a forged signature with tag $\mathbf{t}^* = \mathbf{t}'$. Then, $\mathcal{B}$ simulates the EUF-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 $\mathbf{R}_0 \leftarrow_r (D_{\mathbb{Z}^{\bar{m}}, \omega(\sqrt{\log n})})^{nk}$, 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 $\mathbf{t} \leftarrow_r \{0,1\}^\ell$. 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 [1]. $\mathcal{B}$ 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 2 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 $\mathbf{R}_0 \leftarrow_r (D_{\mathbb{Z}^{\bar{m}},\omega(\sqrt{\log n})})^{nk}$, 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 3. 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 $Q = \mathrm{poly}(n)$ 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 $\mathbf{t} \leftarrow_r \{0,1\}^\ell$ 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 $i \in \{1, \ldots, u\}$ and $\mathbf{S}_{M^*} = \mathbf{0}$ (using the fact that $M^* \notin \{M_1, \ldots, M_u\}$) 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-CMA security of $\mathcal{SIG}_2$ with probability ϵ in the real game, then it will also output a valid forgery $(\mathbf{M}^*, \mathbf{e}^*)$ 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 2 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 Identity-Based Encryptions from Lattice-based PHFs

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 $C \leftarrow \mathsf{Enc}(mpk, id, M)$.

The standard semantic security of IBE was first introduced in [12]. In this paper, we use the notion called indistinguishable from random in [1], which captures both semantic security and recipient anonymity by requiring the challenge ciphertext to be indistinguishable from a uniformly random element in the ciphertext space. 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 times for different identities.

Challenge. The adversary $\mathcal{A}$ outputs a challenge plaintext M^* and a challenge identity id^* with a restriction that id^* is not used in the user private key query in phase 1. The challenger $\mathcal{C}$ chooses a uniformly random ciphertext C_0 from the ciphertext space. Then, it computes $C_1 \leftarrow \mathsf{Enc}(mpk, id^*, M^*)$. Finally, it randomly chooses a bit $b^* \leftarrow_r \{0,1\}$, and 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 $\mathrm{Adv}^{\text{indr-id-cpa}}_{\mathcal{IBE},\mathcal{A}}(\kappa) = |\Pr[b = b^*] - \frac{1}{2}|$.

Definition 7 (INDr-ID-CPA Security). *We say an IBE scheme $\mathcal{IBE}$ is* INDr-ID-CPA *secure if for any PPT adversary $\mathcal{A}$, its advantage* $\mathrm{Adv}^{\text{indr-id-cpa}}_{\mathcal{IBE},\mathcal{A}}(\kappa)$ *is negligible in κ.*

In the security game against chosen ciphertext attacks (i.e., INDr-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 7 is denoted as INDr-sID-CPA.

5.1 An Identity-Based Encryption with Short Master Public Key

Let integers n, m', v, β, q be polynomials in the security parameter κ, and let $k = \lceil \log_2 q \rceil$. Let $\mathcal{H} = (\mathcal{H}.\mathrm{Gen}, \mathcal{H}.\mathrm{Eval})$ be any $(1, v, \beta)$-PHF with high min-entropy from $\{0,1\}^n$ to $\mathbb{Z}_q^{n \times m'}$. Let $\mathcal{H}.\mathrm{TrapGen}$ and $\mathcal{H}.\mathrm{TrapEval}$ be a pair of trapdoor generation and trapdoor evaluation algorithm of $\mathcal{H}$ that satisfies the conditions in Definition 5. For convenience, we set both the user identity space and the message space as $\{0,1\}^n$. Let integers $\bar{m} = O(n \log q), m = \bar{m} + m'$, $\alpha \in \mathbb{R}$, and large enough $s > \max(\beta, \sqrt{m}) \cdot \omega(\sqrt{\log n})$ 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, 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 $\mathbf{U} \leftarrow_r \mathbb{Z}_q^{n \times n}$, and compute $K \leftarrow \mathcal{H}.\mathrm{Gen}(1^\kappa)$. Finally, return $(mpk, msk) = ((\mathbf{A}, K, \mathbf{U}), \mathbf{R})$.

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

$\mathsf{Enc}(mpk, id \in \{0,1\}^n, M \in \{0,1\}^n)$**:** Given mpk, id and plaintext M, compute the matrix $\mathbf{A}_{id} = (\mathbf{A} \| \mathrm{H}_K(id)) \in \mathbb{Z}_q^{n \times m}$, where $\mathrm{H}_K(id) = \mathcal{H}.\mathrm{Eval}(K, id) \in \mathbb{Z}_q^{n \times m'}$. Then, randomly choose $\mathbf{s} \leftarrow_r \mathbb{Z}_q^n, \mathbf{x}_0 \leftarrow_r D_{\mathbb{Z}^n, \alpha q}, \mathbf{x}_1 \leftarrow_r D_{\mathbb{Z}^{\bar{m}}, \alpha q}$, and compute $(K', td) \leftarrow \mathcal{H}.\mathsf{TrapGen}(1^\kappa, \mathbf{A}, \mathbf{B})$ for some trapdoor matrix $\mathbf{B} \in \mathbb{Z}_q^{n \times m'}$, $(\mathbf{R}_{id}, \mathbf{S}_{id}) = \mathcal{H}.\mathsf{TrapEval}(td, K', id)$. Finally, compute and return the ciphertext $\mathbf{C} = (\mathbf{c}_0, \mathbf{c}_1)$, where

$$\mathbf{c}_0 = \mathbf{U}^t\mathbf{s} + \mathbf{x}_0 + \frac{q}{2}M, \qquad \mathbf{c}_1 = \mathbf{A}_{id}^t\mathbf{s} + \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{R}_{id}^t\mathbf{x}_1 \end{pmatrix} = \begin{pmatrix} \mathbf{A}^t\mathbf{s} + \mathbf{x}_1 \\ \mathrm{H}_K(id)^t\mathbf{s} + \mathbf{R}_{id}^t\mathbf{x}_1 \end{pmatrix}.$$

$\mathsf{Dec}(sk_{id}, \mathbf{C})$**:** Given $sk_{id} = \mathbf{E}_{id}$ and a ciphertext $\mathbf{C} = (\mathbf{c}_0, \mathbf{c}_1)$ under identity id, compute $\mathbf{b} = \mathbf{c}_0 - \mathbf{E}_{id}^t \mathbf{c}_1 \in \mathbb{Z}_q^n$. Then, treat each coordinate of $\mathbf{b} = (b_1, \ldots, b_n)^t$ as an integer in $\mathbb{Z}$, and set $M_i = 1$ if $|b_i - \lfloor \frac{q}{2} \rfloor| \leq \lfloor \frac{q}{4} \rfloor$, else $M_i = 0$, where $i \in \{1, \ldots, n\}$. Finally, return the plaintext $M = (M_0, \ldots, M_n)^t$.

By Proposition 1, we have that $s_1(\mathbf{R}) \leq O(\sqrt{m}) \cdot \omega(\sqrt{\log n})$. For large enough $s \geq \sqrt{m} \cdot \omega(\sqrt{\log n})$, by the correctness of $\mathsf{SampleD}$ we know that $\mathbf{A}_{id}\mathbf{E}_{id} = \mathbf{U}$ and $\|\mathbf{E}_{id}\| \leq s\sqrt{m}$ hold with overwhelming probability. In this case, $\mathbf{c}_0 - \mathbf{E}_{id}^t \mathbf{c}_1 = \mathbf{c}_0 - \mathbf{E}_{id}^t (\mathbf{A}_{id}^t \mathbf{s} + \hat{\mathbf{x}}) = \mathbf{c}_0 - \mathbf{U}^t \mathbf{s} - \mathbf{E}_{id}^t \hat{\mathbf{x}} = \frac{q}{2} M + \mathbf{x}_0 - \mathbf{E}_{id}^t \hat{\mathbf{x}}$, where $\hat{\mathbf{x}} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{R}_X^t \mathbf{x}_1 \end{pmatrix}$. Now, we estimate the size of $\|\mathbf{x}_0 - \mathbf{E}_{id}^t \hat{\mathbf{x}}\|_\infty$. Since $\mathbf{x}_0 \leftarrow_r D_{\mathbb{Z}^n, \alpha q}, \mathbf{x}_1 \leftarrow_r D_{\mathbb{Z}^{\bar{m}}, \alpha q}$, we have that $\|\mathbf{x}_0\|, \|\mathbf{x}_1\| \leq \alpha q \sqrt{m}$ holds with overwhelming probability by Lemma 1. In addition, using the fact that $s_1(\mathbf{R}_X) \leq \beta$, we have that $\|\hat{\mathbf{x}}\| \leq \alpha q \sqrt{m(\beta^2 + 1)}$. Thus, we have that $\|\mathbf{E}_{id}^t \hat{\mathbf{x}}\|_\infty \leq \alpha q m s \sqrt{\beta^2 + 1}$, and $\|\mathbf{x}_0 - \mathbf{E}_{id}^t \hat{\mathbf{x}}\|_\infty \leq 2\alpha q m s \sqrt{\beta^2 + 1}$. This means that the decryption algorithm is correct if we set parameters such that $2\alpha q m s \sqrt{\beta^2 + 1} < \frac{q}{4}$ holds. For instance, we can set the parameters as follows: $m = 4n^{1+\psi}, s = \beta \cdot \omega(\sqrt{\log n}), q = \beta^2 m^2 \cdot \omega(\sqrt{\log n}), \alpha = (\beta^2 m^{1.5} \cdot \omega(\sqrt{\log n}))^{-1}$, where real $\psi \in \mathbb{R}$ satisfies $\log q < n^\psi$.

Under the LWE assumption, our generic IBE scheme $\mathcal{IBE}$ is IND$_r$-ID-CPA secure in the standard model.

Theorem 8. *Let $n, q, m' \in \mathbb{Z}$ and $\alpha, \beta \in \mathbb{R}$ be polynomials in the security parameter κ. For large enough $v = \mathrm{poly}(n)$, let $\mathcal{H} = (\mathcal{H}.\mathrm{Gen}, \mathcal{H}.\mathrm{Eval})$ be any $(1, v, \beta, \gamma, \delta)$-PHF with high min-entropy from $\{0,1\}^n$ to $\mathbb{Z}_q^{n \times m'}$, where $\gamma = \mathrm{negl}(\kappa)$ and $\delta > 0$ is noticeable. Then, if there exists a PPT adversary $\mathcal{A}$ breaking the* IND$_r$-ID-CPA security of $\mathcal{IBE}$ with non-negligible advantage ϵ and making at most $Q < v$ user private key queries, there exists an algorithm $\mathcal{B}$ solving the $\mathrm{LWE}_{q,\alpha}$ problem with advantage at least $\epsilon' \geq \epsilon\delta/3 - \mathrm{negl}(\kappa)$.

The proof is very similar to that in [1]. We defer it to Appendix B for convenience. Actually, by instantiating $\mathcal{H}$ in the generic scheme $\mathcal{IBE}$ with the Type-I PHF construction, we recover the fully secure IBE scheme due to Agrawal et al. [1]. Besides, if $\mathcal{H}$ is replaced by a weak $(1, v, \beta)$-PHF with high min-entropy, we can further show that the resulting scheme is IND$_r$-sID-CPA secure, and subsumes the selectively secure IBE scheme in [1]. Formally,

Corollary 2. *Let $n, m', q \in \mathbb{Z}$ and $\alpha, \beta \in \mathbb{R}$ be polynomials in the security parameter κ. For large enough $v = \mathrm{poly}(n)$, let $\mathcal{H} = (\mathcal{H}.\mathrm{Gen}, \mathcal{H}.\mathrm{Eval})$ be any weak $(1, v, \beta, \gamma, \delta)$-PHF with high min-entropy from $\{0,1\}^n$ to $\mathbb{Z}_q^{n \times m'}$, where $\gamma = \mathrm{negl}(\kappa)$ and $\delta > 0$ is noticeable. Then, under the $\mathrm{LWE}_{q,\alpha}$ assumption, the generic IBE scheme $\mathcal{IBE}$ is* IND$_r$-sID-CPA *secure.*

By instantiating the generic IBE scheme $\mathcal{IBE}$ with our efficient Type-II PHF in Definition 4, we can obtain a fully secure 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$_r$-ID-CPA *security of $\mathcal{IBE}_1$ with non-negligible advantage ϵ and making at most $Q = \mathrm{poly}(\kappa)$ user private key queries, then there exists an algorithm $\mathcal{B}$ solving the $\mathrm{LWE}_{q,\alpha}$ problem with advantage at least $\epsilon' \geq \frac{\epsilon}{48nQ^2} - \mathrm{negl}(\kappa)$.*

Remark 2. Since our Type-II $(1, v, \beta)$-PHF depends on the parameter v in several aspects, the instantiated IBE scheme $\mathcal{IBE}_1$ relies on the particular number Q of user private key queries (because of $Q \leq v$) in terms of the master public key size and the reduction loss. On the first hand, the size of the master public key only depends on Q in a (somewhat) weak sense: for any polynomial Q it only affects the constant factor hidden in the number $O(\log n)$ of matrices in the master public key. When implementing the IBE scheme, one can either prior determine the target security level (or the maximum number Q of allowed user private key queries) before the

setup phase, or set a super polynomial v to generate the master public keys. For example, for $v = n^{\log(\log n)}$, the master public key only contains $O(\log(\log n)\log n)$ matrices, which is still much smaller than the linear function $O(n)$ as that in [1,16]. On the other hand, the reduction loss of $\mathcal{IBE}_1$ also depends on Q (due to our proof of Theorem 3). Unlike the signature scheme $\mathcal{SIG}_2$, it is unclear if one can reduce the reduction loss with some modifications/improvements. Besides, it is also interesting to investigate the possibility of giving a proof of Theorem 3 with an improved $\delta > 0$.

5.2 Extensions

Hierarchical IBE. Using the trapdoor delegation techniques in [1,16,45], 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 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 $\mathbf{A}_{id} = (\mathbf{A}\|\mathrm{H}_{K_1}(id_1)\|\cdots\|\mathrm{H}_{K_{d'}}(id_{d'}))$. 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 [1,16,45]. We note that as previous schemes using similar partitioning techniques [1,16], 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 [53] 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 INDr-ID-CCA secure IBE scheme by using a two-level INDr-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 [1,16,45]. 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 $\mathbf{A}_{id}$.

6 Conclusions and Open Problems

We introduced the notion of lattice-based PHFs and mainly gave two 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 [14,45]. By instantiating the generic scheme with our efficient lattice-based PHF constructions, we immediately obtained a lattice-based short signature scheme with short verification keys. Furthermore, we showed how to combine two concrete lattice-based PHFs to construct a short signature scheme from

weaker assumptions. We also constructed a generic construction of IBE scheme from lattice-based PHFs with an enhanced property called high min-entropy. Our generic scheme subsumed the IBE schemes in [1]. By instantiating the generic IBE scheme with our efficient Type-II lattice-based PHF construction, we obtained a fully secure IBE scheme with short master public key 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 would like to thank Eike Kiltz and Xusheng Zhang for their helpful discussions. We also thank the anonymous reviewers of Crypto 2016 for their insightful advices. Jiang Zhang and Zhenfeng Zhang are supported by the National Grand Fundamental Research (973) Program of China under Grant No. 2013CB338003 and the National Natural Science Foundation of China under Grant No. U1536205. Yu Chen is supported by the National Natural Science Foundation of China under Grant Nos. 61303257 and 61379141, and by the State Key Laboratory of Cryptologys Open Project under Grant No. MMKFKT201511.

References

1. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: Gilbert, H. (ed.) EUROCRYPT 2010, LNCS, vol. 6110, pp. 553–572. Springer (2010)
2. 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)
3. Agrawal, S., Freeman, D., Vaikuntanathan, V.: Functional encryption for inner product predicates from learning with errors. In: Lee, D., Wang, X. (eds.) ASIACRYPT 2011, LNCS, vol. 7073, pp. 21–40. Springer (2011)
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. Apon, D., Fan, X., Liu, F.H.: Fully-secure lattice-based ibe as compact as pke. Cryptology ePrint Archive, Report 2016/125 (2016)
8. 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)
9. 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)
10. 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)
11. 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)
12. 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)
13. 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)

14. 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)
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., Stehl, 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. Katz, J.: Digital Signatures. Springer (2010)
39. Krawczyk, H., Rabin, T.: Chameleon signatures. In: NDSS 2000.
40. Kumar, R., Rajagopalan, S., Sahai, A.: Coding constructions for blacklisting problems without computational assumptions. In: CRYPTO '99. pp. 609–623. Springer (1999)
41. Lyubashevsky, V.: Lattice signatures without trapdoors. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012, LNCS, vol. 7237, pp. 738–755. Springer (2012)
42. 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)
43. 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)
44. Lyubashevsky, V., Peikert, C., Regev, O.: A toolkit for ring-LWE cryptography. In: Johansson, T., Nguyen, P. (eds.) EUROCRYPT 2013, LNCS, vol. 7881, pp. 35–54. Springer (2013)
45. 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)
46. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on gaussian measures. SIAM J. Comput. 37, 267–302 (April 2007)
47. 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)
48. 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)
49. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC 2005. pp. 84–93. ACM (2005)
50. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G., Chaum, D. (eds.) CRYPTO '84, LNCS, vol. 196, pp. 47–53. Springer (1984)
51. Vershynin, R.: Introduction to the non-asymptotic analysis of random matrices. arXiv preprint arXiv:1011.3027 (2010)
52. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005, LNCS, vol. 3494, pp. 114–127. Springer (2005)
53. 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)
54. Yamada, S.: Adaptively secure identity-based encryption from lattices with asymptotically shorter public parameters. In: Fischlin, M., Coron, J.S. (eds.) EUROCRYPT 2016, LNCS, vol. 9666, pp. 32–62. Springer (2016)
55. 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)
56. 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)

A Generalized Leftover Hash Lemma

In this section, we recall the definition of universal hash function and the generalized leftover hash lemma [23]. Formally, a family of hash functions $\mathcal{H} = \{\mathrm{H}_K : \mathcal{X} \to \mathcal{Y}\}_{K \in \mathcal{K}}$ with key space $\mathcal{K}$ is universal if for all $X_1 \neq X_2$, we have $\Pr_{K \leftarrow_r \mathcal{K}}[H_K(X_1) = H_K(X_2)] = 1/|\mathcal{Y}|$. Let n, m be any positive integer, and q be a prime. It is well known that $\mathcal{H} = \{\mathrm{H}_{\mathbf{A}} : \mathbb{Z}_q^m \to \mathbb{Z}_q^n\}_{\mathbf{A} \in \mathbb{Z}_q^{n \times m}}$ defined by $\mathrm{H}_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x}$ is a family of universal hash functions. Besides, for any random variable X taking values in $\mathcal{X}$, the guessing probability of X is defined as $\gamma(X) = \max_{\hat{X} \in \mathcal{X}} \Pr[X = \hat{X}]$.

Lemma 5 (Generalized Leftover Hash Lemma [23]). *If $\mathcal{H} = \{\mathrm{H}_K : \mathcal{X} \to \mathcal{Y}\}_{K \in \mathcal{K}}$ is a family of universal hash functions. Let $f : \mathcal{X} \to \mathcal{Z}$ be a function. Then, for any random variables X over $\mathcal{X}$, the statistical difference between $(H, H(X), f(X))$ and $(H, U, f(X))$ is at most $\frac{1}{2} \cdot \sqrt{\gamma(X) \cdot |\mathcal{Y}| \cdot |\mathcal{Z}|}$, where H and U are uniformly distributed over $\mathcal{H}$ and $\mathcal{Y}$, respectively.*

B Proof of Theorem 8

We begin by first restating Theorem 8 for convenience.

Theorem 8. *Let $n, q, m' \in \mathbb{Z}$ and $\alpha, \beta \in \mathbb{R}$ be polynomials in the security parameter κ. For large enough $v = \mathrm{poly}(n)$, let $\mathcal{H} = (\mathcal{H}.\mathrm{Gen}, \mathcal{H}.\mathrm{Eval})$ be any $(1, v, \beta, \gamma, \delta)$-PHF with high min-entropy from $\{0,1\}^n$ to $\mathbb{Z}_q^{n \times m'}$, where $\gamma = \mathrm{negl}(\kappa)$ and $\delta > 0$ is noticeable. Then, if there exists a PPT adversary $\mathcal{A}$ breaking the* INDr-ID-CPA security of $\mathcal{IBE}$ with non-negligible advantage ϵ and making at most $Q < v$ user private key queries, there exists an algorithm $\mathcal{B}$ solving the $\mathrm{LWE}_{q,\alpha}$ problem with advantage at least $\epsilon' \geq \epsilon\delta/3 - \mathrm{negl}(\kappa)$.

The proof is very similar to that in [1]. We defer it to Appendix B for lack of space. Actually, by instantiating $\mathcal{H}$ in the generic scheme $\mathcal{IBE}$ with the Type-I PHF construction, we recover the fully secure IBE scheme due to Agrawal et al. [1]. Besides, if $\mathcal{H}$ is replaced by a weak $(1, v, \beta)$-PHF with high min-entropy, we can further show that the resulting scheme is INDr-sID-CPA secure, and subsumes the selectively secure IBE scheme in [1]. Formally,

Proof. In the following, we use a sequence of games from Game 0 to Game 5. Informally, Game 0 is exactly the real security game as in Definition 7 where the challenger honestly encrypts the challenge plaintext, while Game 5 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 5 with non-negligible advantage, which is contradictory to the fact that Game 5 is a random game. Let $\mathcal{H}.\mathrm{TrapGen}$ and $\mathcal{H}.\mathrm{TrapEval}$ be a pair of trapdoor generation and trapdoor evaluation algorithm of $\mathcal{H}$ that satisfies the conditions in Definition 5. For simplicity, we fix the trapdoor matrix $\mathbf{B} = \mathbf{G} \in \mathbb{Z}_q^{n \times nk}$ throughout the proof. One can extend the proof to any other general trapdoor matrix $\mathbf{B}$ that allows to efficiently sample short vector $\mathbf{v}$ satisfying $\mathbf{Bv} = \mathbf{u}$ for any $\mathbf{u} \in \mathbb{Z}_q^n$, by using the trapdoor delegation techniques in [1].

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

Setup. First compute $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^n, 1^{\bar{m}}, q, \mathbf{I}_n)$ such that $\mathbf{A} \in \mathbb{Z}_q^{n \times \bar{m}}$, $\mathbf{R} = \mathbb{Z}_q^{(\bar{m}-nk) \times nk}$. Then, randomly choose $\mathbf{U} \leftarrow_r \mathbb{Z}_q^{n \times n}$, and compute $K \leftarrow \mathcal{H}.\mathrm{Gen}(1^\kappa)$. 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 $id \in \{0,1\}^n$, compute the hash value $\mathbf{A}_{id} = (\mathbf{A} \| \mathrm{H}_K(id)) \in \mathbb{Z}_q^{n \times m}$, where $\mathrm{H}_K(id) = \mathcal{H}.\mathrm{Eval}(K, id) \in \mathbb{Z}_q^{n \times nk}$. Then, compute $\mathbf{E}_{id} \leftarrow \mathsf{SampleD}(\mathbf{R}, \mathbf{A}_{id}, \mathbf{I}_n, \mathbf{U}, s)$, and send the user private key $sk_{id} = \mathbf{E}_{id} \in \mathbb{Z}^{m \times n}$ to the adversary $\mathcal{A}$.

Challenge. At some time, the adversary $\mathcal{A}$ outputs a challenge identity id^* and a plaintext $M^* \in \{0,1\}^n$ with the restriction that it never obtains the user private key of id^* in Phase 1. The challenger $\mathcal{C}$ first randomly chooses $\mathbf{C}_0 \leftarrow_r \mathbb{Z}_q^n \times \mathbb{Z}_q^m$, $\mathbf{s} \leftarrow_r \mathbb{Z}_q^n$, $\mathbf{x}_0 \leftarrow_r D_{\mathbb{Z}^n, \alpha q}$, and $\mathbf{x}_1 \leftarrow_r D_{\mathbb{Z}^{\bar{m}}, \alpha q}$. Then, it computes $(K', td) \leftarrow \mathcal{H}.\mathsf{TrapGen}(1^\kappa, \mathbf{A}, \mathbf{G})$, $(\mathbf{R}_{id^*}, \mathbf{S}_{id^*}) = \mathcal{H}.\mathsf{TrapEval}(td, K', id^*)$ and sets $\mathbf{C}_1 = (\mathbf{c}_0^*, \mathbf{c}_1^*)$, where

$$\mathbf{c}_0^* = \mathbf{U}^t\mathbf{s} + \mathbf{x}_0 + \frac{q}{2}M_{b^*}, \quad \mathbf{c}_1^* = \mathbf{A}_{id^*}^t\mathbf{s} + \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{R}_{id^*}^t\mathbf{x}_1 \end{pmatrix} = \begin{pmatrix} \mathbf{A}^t\mathbf{s} + \mathbf{x}_1 \\ \mathrm{H}_K(id^*)^t\mathbf{s} + \mathbf{R}_{id^*}^t\mathbf{x}_1 \end{pmatrix},$$

where $\mathbf{A}_{id^*} = (\mathbf{A} \| \mathrm{H}_K(id^*)) \in \mathbb{Z}_q^{n \times m}$ and $\mathrm{H}_K(id^*) = \mathcal{H}.\mathrm{Eval}(K, id^*) \in \mathbb{Z}_q^{n \times nk}$. Finally, it randomly chooses a bit $b^* \leftarrow_r \{0,1\}$, and 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 be the event that $\mathcal{C}$ outputs 1 in Game i for $i \in \{0,1,\ldots,5\}$.

Lemma 6. $|\Pr[F_0] - \frac{1}{2}| = \epsilon$.

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 the challenger $\mathcal{C}$ changes the setup and the challenge phases as follows.

Setup. First compute $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^n, 1^{\bar{m}}, q, \mathbf{I}_n)$ such that $\mathbf{A} \in \mathbb{Z}_q^{n \times \bar{m}}$, $\mathbf{R} = \mathbb{Z}_q^{(\bar{m}-nk)\times nk}$. Then, randomly choose $\mathbf{U} \leftarrow_r \mathbb{Z}_q^{n\times n}$, and compute $(K', td) \leftarrow \mathcal{H}.\mathrm{TrapGen}(1^\kappa, \mathbf{A}, \mathbf{G})$. 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.

Challenge. This phase is the same as in Game 2 except that the challenger $\mathcal{C}$ directly uses the pair (K', td) produced in the setup phase to generate the ciphertext $\mathbf{C}_1 = (\mathbf{c}_0^*, \mathbf{c}_1^*)$.

Lemma 7. *If $\mathcal{H}$ is a PHF with high min-entropy, then* $|\Pr[F_1] - \Pr[F_0]| \leq \mathrm{negl}(\kappa)$.

Proof. By the first condition of high min-entropy in Definition 5, for any $K \leftarrow_r \mathcal{H}.\mathrm{Gen}(1^\kappa)$, $(K', td) \leftarrow \mathcal{H}.\mathrm{TrapGen}(1^\kappa, \mathbf{A}, \mathbf{G})$, any $id^* \in \{0,1\}^n$ and any $\mathbf{x}_1 \in \mathbb{Z}_q^{\bar{m}}$, we have that the statistical distance between $(\mathbf{A}, K', \mathbf{R}_{id^*}^t \mathbf{x}_1)$ and $(\mathbf{A}, K, \mathbf{R}_{id^*}^t \mathbf{x}_1)$ is negligible, where $(\mathbf{R}_{id^*}, \mathbf{S}_{id^*}) = \mathcal{H}.\mathsf{TrapEval}(td, K', id^*)$. This means that the master public key mpk and the ciphertext $\mathbf{C}_1 = (\mathbf{c}_0^*, \mathbf{c}_1^*)$ in Game 1 are statistically close to that in Game 0. Thus, we have $|\Pr[F_1] - \Pr[F_0]| \leq \mathrm{negl}(\kappa)$. □

Game 2. This game is identical to Game 1 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 $\mathcal{C}$ first defines the following function

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

where $(\hat{\mathbf{R}}_{id^*}, \hat{\mathbf{S}}_{id^*}) = \mathcal{H}.\mathrm{TrapEval}(\hat{td}, \hat{K}, id^*)$ and $(\hat{\mathbf{R}}_{id_i}, \hat{\mathbf{S}}_{id_i}) = \mathcal{H}.\mathrm{TrapEval}\ (\hat{td}, \hat{K}, id_i)$. Then, $\mathcal{C}$ proceeds the following steps:

1. **Abort check**: Let (td, K') be produced in the setup phase when generating the master public key $mpk = (\mathbf{A}, K', \mathbf{U})$, the challenger $\mathcal{C}$ computes the value of $\tau(td, K', I^*)$. 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 $p = \Pr[\tau(\hat{td}, \hat{K}, I^*) = 0]$ over the random choice of $(\hat{td}, \hat{K})$. Then, $\mathcal{C}$ samples $O(\epsilon^2 \log(\epsilon^{-1})\delta^{-1}\log(\delta^{-1}))$ times the probability p by independently running $(\hat{td}, \hat{K}) \leftarrow \mathcal{H}.\mathrm{TrapGen}(1^\kappa, \mathbf{A}, \mathbf{G})$ and evaluating $\tau(\hat{td}, \hat{K}, I^*)$ 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 3. As in [52,9,1,16,28], 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 [52], 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 [9] showed that the artificial abort can be avoided. Because our construction uses general lattice-based PHFs as a "black-box", 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 [52].

For $i \in \{2, 3, 4, 5\}$, let $\tilde{p}_i$ be the probability that $\mathcal{C}$ does not abort in the abort check stage in Game i, and let p_i be the probability in the artificial abort stage of Game i defined by $p_i = \Pr[\tau(\hat{td}, \hat{K}, I^*) = 0]$. Since the adversary might obtain some information of td from the challenge ciphertext $\mathbf{C}_{b^*}$, the probability $\tilde{p}_i$ might not be equal to the probability p. However, we will show later that the two probabilities can be very close under the LWE assumption. Formally, let Γ_i be the absolute difference between $\tilde{p}_i$ and p_i (i.e., $\Gamma_i = |\tilde{p}_i - p_i|$), we have the following lemma.

Lemma 8. *If $\mathcal{H}$ is a $(1, v, \beta, \gamma, \delta)$-PHF and $Q \leq v$, then $|\Pr[F_2] - \frac{1}{2}| \geq \frac{1}{2}\epsilon(\delta - \Gamma_2)$.*

So as not to interrupt the game sequences, we defer the proof of Lemma 8 to the end of the Game 5.

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 $id \in \{0,1\}^n$, first compute $\mathbf{A}_{id} = (\mathbf{A} \| \mathrm{H}_{K'}(id)) \in \mathbb{Z}_q^{n \times m}$, where $\mathrm{H}_{K'}(id) = \mathcal{H}.\mathrm{Eval}(K', id) \in \mathbb{Z}_q^{n \times nk}$. Then, compute $(\mathbf{R}_{id}, \mathbf{S}_{id}) = \mathcal{H}.\mathrm{TrapEval}(td, K', id)$. If $\mathbf{S}_{id}$ is not invertible, the challenger $\mathcal{C}$ outputs a uniformly random bit and aborts the game. Otherwise, compute $\mathbf{E}_{id} \leftarrow \mathsf{SampleD}(\mathbf{R}_{id}, \mathbf{A}_{id}, \mathbf{S}_{id}, \mathbf{U}, s)$, and send $sk_{id} = \mathbf{E}_{id} \in \mathbb{Z}^{m \times n}$ 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 $(\mathbf{R}_{id^*}, \mathbf{S}_{id^*}) = \mathcal{H}.\mathrm{TrapEval}(td, K', id^*)$. Note that if $\mathbf{S}_{id^*} = \mathbf{0}$, we have $\mathrm{H}_{K'}(id^*) = \mathbf{A}\mathbf{R}_{id^*}$ and $\mathbf{c}_1^* = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{R}_{id^*}^t \mathbf{b}_1 \end{pmatrix}$ for some vector $\mathbf{b}_1 = \mathbf{A}^t\mathbf{s} + \mathbf{x}_1 \in \mathbb{Z}_q^{\bar{m}}$.

[10] In general, the sampling procedure would make 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.

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 9. *If $\mathcal{H}$ is a $(1, v, \beta, \gamma, \delta)$-PHF and $Q \leq v$, then $\Pr[F_3] = \Pr[F_2]$ and $\Gamma_3 = \Gamma_2$.*

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 $\mathsf{SampleD}$ algorithm to successfully generate the user private keys by the fact that $s_1(\mathbf{R}_{id}) \leq \beta$ and $s > \max(\beta, \sqrt{m}) \cdot \omega(\sqrt{\log n})$. Thus, if $\mathcal{C}$ does not abort during the game, then Game 3 is identical to Game 2 in the adversary $\mathcal{A}$'s view. In all, we have that $\Pr[F_4] = \Pr[F_3]$ and $\Gamma_3 = \Gamma_2$ hold. □

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 $\mathbf{A} \leftarrow_r \mathbb{Z}_q^{n \times \bar{m}}$, $\mathbf{U} \leftarrow_r \mathbb{Z}_q^{n \times n}$, and compute the trapdoor mode key $(K', td) \leftarrow \mathcal{H}.\mathrm{TrapGen}(1^\kappa, \mathbf{A}, \mathbf{G})$. 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 $\mathbf{C}_1 = (\mathbf{c}_0^*, \mathbf{c}_1^*)$ as follows: randomly choose vector $\mathbf{b}_0 \leftarrow_r \mathbb{Z}_q^n, \mathbf{b}_1 \leftarrow_r \mathbb{Z}_q^{\bar{m}}$, and compute

$$\mathbf{c}_0^* = \mathbf{b}_0 + \frac{q}{2} M_{b^*}, \qquad \mathbf{c}_1^* = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{R}_{id^*}^t \mathbf{b}_1 \end{pmatrix},$$

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

Lemma 10. *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 $|\Pr[F_4] - \Pr[F_3]| \leq \epsilon'$ and $|\Gamma_4 - \Gamma_3| \leq \epsilon'$ hold.*

Proof. We construct an algorithm $\mathcal{B}$ for the $\mathrm{LWE}_{q,\alpha}$ as follows. Given the $\mathrm{LWE}_{q,\alpha}$ challenge instance $(\hat{\mathbf{U}}, \hat{\mathbf{b}}_0) \in \mathbb{Z}_q^{n \times n} \times \mathbb{Z}_q^n$ and $(\hat{\mathbf{A}}, \hat{\mathbf{b}}_1) \in \mathbb{Z}_q^{n \times \bar{m}} \times \mathbb{Z}_q^{\bar{m}}$. $\mathcal{B}$ 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 $(\hat{\mathbf{U}}, \hat{\mathbf{b}}_0) \in \mathbb{Z}_q^{n \times n} \times \mathbb{Z}_q^n$ and $(\hat{\mathbf{A}}, \hat{\mathbf{b}}_1) \in \mathbb{Z}_q^{n \times \bar{m}} \times \mathbb{Z}_q^{\bar{m}}$ 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 $|\Pr[F_4] - \Pr[F_3]| \leq \epsilon'$ and $|\Gamma_4 - \Gamma_3| \leq \epsilon'$ hold by our assumption. □

Game 5. This game is identical to Game 4 except that the challenger $\mathcal{C}$ makes the following changes.

Setup. First compute $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^n, 1^{\bar{m}}, q, \mathbf{I}_n)$ such that $\mathbf{A} \in \mathbb{Z}_q^{n \times \bar{m}}$, $\mathbf{R} = \mathbb{Z}_q^{(\bar{m}-nk) \times nk}$. Then, the challenger $\mathcal{C}$ randomly chooses matrix $\mathbf{U} \leftarrow_r \mathbb{Z}_q^{n \times n}$, computes $K \leftarrow_r \mathcal{H}.\mathrm{Gen}(1^\kappa)$ and $(K', td) \leftarrow_r \mathcal{H}.\mathrm{TrapGen}(1^\kappa, \mathbf{A}, \mathbf{G})$. Finally, send the master public key $mpk = (\mathbf{A}, K, \mathbf{U})$ to the adversary $\mathcal{A}$, and keep $(\mathbf{R}, K', td)$ private.

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

Challenge. This phase is the same as in Game 4 except that the challenger generates the ciphertext $\mathbf{C}_1 = (\mathbf{c}_0^*, \mathbf{c}_1^*)$ by randomly choosing $\mathbf{c}_0^* \leftarrow_r \mathbb{Z}_q^n$ and $\mathbf{c}_1^* \leftarrow_r \mathbb{Z}_q^m$.

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 11. *If $\mathcal{H}$ is a $(1, v, \beta, \mathrm{negl}(\kappa), \delta)$-PHF with min-entropy, then we have $|\Pr[F_5] - \Pr[F_4]| \leq \mathrm{negl}(\kappa)$ and $|\Gamma_5 - \Gamma_4| \leq \mathrm{negl}(\kappa)$.*

Proof. Since $\mathcal{H}$ is a $(1, v, \beta, \mathrm{negl}(\kappa), \delta)$-PHF, the statistical distance between the master public key mpk in Game 4 and that in Game 5 is negligible by the property of $\mathsf{TrapGen}$. By the property of $\mathsf{SampleD}$, the distribution of user private keys in Game 5 is almost identical to that in Game 4. Since both $\mathbf{b}_0, \mathbf{b}_1$ in Game 4 are uniformly random over $\mathbb{Z}_q^n$ and $\mathbb{Z}_q^{\bar{m}}$, the challenge ciphertext in Game 4 is statistically close to that in Game 5 by the second condition of high min-entropy in Definition 5. Finally, using the fact that both Game 4 and Game 5 implement the same abort strategy in the guess phase, we have that Game 4 is negligibly close to Game 5 in the adversary $\mathcal{A}$'s view. Thus, we have that both $|\Pr[F_5] - \Pr[F_4]| \leq \mathrm{negl}(\kappa)$ and $|\Gamma_5 - \Gamma_4| \leq \mathrm{negl}(\kappa)$ hold. □

Lemma 12. $\Pr[F_5] = \frac{1}{2}$ *and* $\Gamma_5 = 0$.

Proof. The first claim follows from the fact that $\mathbf{C}_1$ is uniformly random. Since both the master pubic key mpk and the challenge ciphertext are independent from the random choice of td in Game 5, the challenger $\mathcal{C}$ can actually compute $(K', td) \leftarrow \mathcal{H}.\mathrm{TrapGen}(1^\kappa, \mathbf{A}, \mathbf{G})$ in the guess phase, and use (K', td) in the abort check stage. By the definition of Γ_5, we have $\Gamma_5 = 0$. This completes the proof. □

By Lemma 8 and Lemma 9, we have $|\Pr[F_3] - \frac{1}{2}| \geq \frac{1}{2}\epsilon(\delta - \Gamma_3)$. By Lemma 11 and Lemma 12, we have $\Pr[F_4] \leq \frac{1}{2} + \mathrm{negl}(\kappa)$ and $\Gamma_4 \leq \mathrm{negl}(\kappa)$. By the fact that and $|\Pr[F_4] - \Pr[F_3]| \leq \epsilon'$ and $|\Gamma_4 - \Gamma_3| \leq \epsilon'$ in Lemma 10, we have $\frac{1}{2}\epsilon(\delta - \epsilon') - \mathrm{negl}(\kappa) \leq |\Pr[F_3] - \frac{1}{2}| \leq \epsilon' - \mathrm{negl}(\kappa)$. This shows that $\epsilon' \geq \frac{\epsilon\delta}{3} - \mathrm{negl}(\kappa)$ holds, which completes the proof of Theorem 8. □

Proof of Lemma 8. Let $\mathcal{QID} = (\{0,1\}^n)^{Q+1}$ 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 $i \in \{1, 2\}$. Let $\mathcal{E}$ be the event that $\mathcal{C}$ aborts in Game 2. 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 2, we have that $\Pr[F_2(I) \wedge \mathcal{E}] = \frac{1}{2}\Pr[\mathcal{Q}(I) \wedge \mathcal{E}]$ and $\Pr[F_2(I) \wedge \neg\mathcal{E}] = \Pr[F_1(I) \wedge \neg\mathcal{E}] = \Pr[F_1(I)]\Pr[\neg\mathcal{E} | \mathcal{Q}(I)]$ hold. By a simple calculation, we

have

$$\begin{aligned}|\Pr[F_2]-\tfrac{1}{2}| &= |\textstyle\sum_{I\in\mathcal{QID}}(\Pr[F_2(I)\wedge\mathcal{E}]+\Pr[F_2(I)\wedge\neg\mathcal{E}])-\tfrac{1}{2}|\\ &= |\textstyle\sum_{I\in\mathcal{QID}}(\Pr[F_2(I)\wedge\neg\mathcal{E}]-\tfrac{1}{2}\Pr[\mathcal{Q}(I)\wedge\neg\mathcal{E}])|\\ &= |\textstyle\sum_{I\in\mathcal{QID}}(\Pr[F_1(I)]-\tfrac{1}{2}\Pr[\mathcal{Q}(I)])\Pr[\neg\mathcal{E}|\mathcal{Q}(I)]|.\end{aligned}$$

Since $\Pr[F_i(I)]\leq\Pr[Q(I)]$, we have that $|\Pr[F_1(I)-\frac{1}{2}\Pr[\mathcal{Q}(I)]|\leq\frac{1}{2}\Pr[\mathcal{Q}(I)]$ holds. Note that the term $\Pr[F_1(I)-\frac{1}{2}\Pr[\mathcal{Q}(I)]$ can be either positive or negative. Let $\mathcal{QID}^+$ (resp. $\mathcal{QID}^-$) be the set of identities such that $\Pr[F_1(I)-\frac{1}{2}\Pr[\mathcal{Q}(I)]$ is positive (resp. negative), and let $\eta(I)=\Pr[\neg\mathcal{E}|\mathcal{Q}(I)]$. In addition, let $\eta_{max}=\max_{I\in\mathcal{QID}}\eta(I)$ and $\eta_{min}=\min_{I\in\mathcal{QID}}\eta(I)$. Then, we have

$$\begin{aligned}|\Pr[F_2]-\tfrac{1}{2}| &= |\textstyle\sum_{I\in\mathcal{QID}^+}(\Pr[F_1(I)]-\tfrac{1}{2}\Pr[\mathcal{Q}(I)])\eta(I)\\ &\quad+\textstyle\sum_{I\in\mathcal{QID}^-}(\Pr[F_1(I)]-\tfrac{1}{2}\Pr[\mathcal{Q}(I)])\eta(I)|\\ &\geq \eta_{min}|\textstyle\sum_{I\in\mathcal{QID}}(\Pr[F_1(I)]-\tfrac{1}{2}\Pr[\mathcal{Q}(I)])|-\tfrac{1}{2}(\eta_{max}-\eta_{min})\\ &= \eta_{min}|\Pr[F_1]-\tfrac{1}{2}|-\tfrac{1}{2}(\eta_{max}-\eta_{min}).\end{aligned}$$

By the definition of $\tilde{p}_2$ and p_2 in Game 2, we have $\eta(I)=\Pr[\neg\mathcal{E}|\mathcal{Q}(I)]=\tilde{p}_2\frac{\delta}{p'}$, where p' is an estimate of p_2. Since the challenger $\mathcal{C}$ always samples $O(\epsilon^2\log(\epsilon^{-1})\delta^{-1}\log(\delta^{-1}))$ times the probability p_2 to compute p', we have that $\Pr[p'>p_2(1+\frac{\epsilon}{8})]<\delta\frac{\epsilon}{8}$ and $\Pr[p'<p_2(1-\frac{\epsilon}{8})]<\delta\frac{\epsilon}{8}$ hold by the Chernoff bounds. Then, we have

$$\begin{aligned}&\eta_{max}\leq(1-\delta\tfrac{\epsilon}{8})\tilde{p}_2\tfrac{\delta}{p_2(1-\frac{\epsilon}{8})}\\ &\eta_{min}\geq(1-\delta\tfrac{\epsilon}{8})\tilde{p}_2\tfrac{\delta}{p_2(1+\frac{\epsilon}{8})}\geq\tfrac{7\delta\tilde{p}_2}{9p_2}\\ &\eta_{max}-\eta_{min}\leq(1-\delta\tfrac{\epsilon}{8})\tfrac{\epsilon\delta\tilde{p}_2}{4(1-\frac{\epsilon^2}{64})p_2}\leq\tfrac{16\epsilon\delta\tilde{p}_2}{63p_2}\end{aligned}$$

By Lemma 6 and Lemma 7, $|\Pr[F_1]-\frac{1}{2}|\geq\epsilon-\mathrm{negl}(\kappa)$ holds. Then, we have

$$\begin{aligned}|\Pr[F_2]-\tfrac{1}{2}| &\geq \eta_{min}|\Pr[F_1]-\tfrac{1}{2}|-\tfrac{1}{2}(\eta_{max}-\eta_{min})\\ &\geq \tfrac{7\delta\tilde{p}_2}{9p_2}(\epsilon-\mathrm{negl}(\kappa))-\tfrac{8\epsilon\delta\tilde{p}_2}{63p_2}\\ &\geq \tfrac{\epsilon\delta(p_2-\Gamma_2)}{2p_2}\geq\tfrac{1}{2}\epsilon(\delta-\Gamma_2),\end{aligned}$$

where the last two inequalities are due to the fact that $\Gamma_2=|\tilde{p}_2-p_2|$ and $p_2\geq\delta$. This completes the proof of Lemma 8. □