Almost Tight Security in Lattices with Polynomial Moduli – PRF, IBE, All-but-many LTF, and More

Qiqi Lai[1(✉)], Feng-Hao Liu[2(✉)], and Zhedong Wang[2(✉)]

[1] School of Computer Science, Shaanxi Normal University, Xi'an, Shaanxi, China
laiqq@snnu.edu.cn
[2] Florida Atlantic University, Boca Raton, FL, USA
{fenghao.liu,wangz}@fau.edu

Abstract. Achieving tight security is a fundamental task in cryptography. While one of the most important purposes of this task is to improve the overall efficiency of a construction (by allowing smaller security parameters), many current lattice-based instantiations do not completely achieve the goal. Particularly, a super-polynomial modulus seems to be necessary in all prior work for (almost) tight schemes that allow the adversary to conduct queries, such as PRF, IBE, and Signatures. As the super-polynomial modulus would affect the noise-to-modulus ratio and thus increase the parameters, this might cancel out the advantages (in efficiency) brought from the tighter analysis. To determine the full power of tight security/analysis in lattices, it is necessary to determine whether the super-polynomial modulus restriction is inherent.

In this work, we remove the super-polynomial modulus restriction for many important primitives – PRF, IBE, All-but-many Lossy Trapdoor Functions, and Signatures. The crux relies on an improvement over the framework of Boyen and Li (Asiacrypt 16), and an almost tight reduction from LWE to LWR, which improves prior work by Alwen et al. (Crypto 13), Bogdanov et al. (TCC 16), and Bai et al. (Asiacrypt 15). By combining these two advances, we are able to derive these almost tight schemes under LWE with a polynomial modulus.

1 Introduction

Tight Security. The reduction framework is a powerful tool to analyze security of a cryptographic construction by relating its security to some suitable mathematical hard problem, such as problems of integer factoring, discrete logs, shortest vector in lattices, and many others [19,35,46]. This framework can be described roughly as follows: assume that there exists a $(t_{\mathcal{A}}, \varepsilon_{\mathcal{A}})$-adversary $\mathcal{A}$ that breaks the cryptographic construction, then we can construct a $(t_{\mathcal{B}}, \varepsilon_{\mathcal{B}})$-reduction algorithm $\mathcal{B}$ that uses $\mathcal{A}$ as a subroutine and solves the underlying hard problem.[1]

[1] We use the notation of (t, ε) to denote an algorithm that breaks a crypto system or solves a hard problem within running time t and with advantage ε.

A. Kiayias et al. (Eds.): PKC 2020, LNCS 12110, pp. 652–681, 2020.
https://doi.org/10.1007/978-3-030-45374-9_22

To evaluate how tight the security of the cryptographic scheme is with respect to the hardness of the underlying problem, we establish analysis of bounds in the form: $\varepsilon_{\mathcal{B}} \geq \varepsilon_{\mathcal{A}}/\theta$ and $t_{\mathcal{B}} \leq kt_{\mathcal{A}} + o(t_{\mathcal{A}})$, and then use $k\theta$ as a measure of tightness – the smaller this quantity is, the tighter the security can achieve. The cryptographic scheme is considered to be (1) *tight* (with respect to the underlying hard problem) if $k\theta = c$ for some constant independent of the adversary, and (2) *almost tight* (with respect to the underlying hard problem) if $k\theta = \mathsf{poly}(\lambda)$ for some small polynomial of the security parameter, independent of the adversary.

Achieving tight security is a meaningful task, particularly when one can prove the same or perhaps slightly less efficient scheme has a tight reduction than a non-tight one. From a theoretical point of view, tightness indicates that security of a crypto scheme is (extremely) closely related to the hardness of the underlying hard problem, which is the optimal case we can expect from the provable security theory. By knowing the (almost) tight relation, we would know how aggressively we can set the security parameter, which is important for practical efficiency.

This subject has drawn a large amount of attention. For symmetric key primitives, we know how to achieve almost tight pseudorandom functions (PRFs) [8,26,41] with respect to various assumptions. Later on, the community turned the focus to public-key primitives. For example, Waters [53] stated an open problem of constructing a tightly, adaptively secure IBE scheme from standard computational hardness assumptions without random oracles. In addition to IBE, progress has been made for various other schemes, including public-key encryption and signature (e.g, [5,10,24,28,32,33]).

Progress in Lattices. While research in this line is active, most results were with respect to assumptions on groups [10,24,33] or integer factorization [9,39]. For other important or post-quantum assumptions such as lattices, only a few results are known even for almost tight security. For symmetric-key primitives, there are only two almost tight PRFs from the learning with error assumption (LWE) [8,41]. For public-key primitives, Boyen and Li [16] constructed the first almost tight IBE based on LWE by using a novel application of (key) homomorphic evaluation of PRF. Later in subsequent work, Boyen and Li [17], and Libert et al. [41] generalized this technique to construct almost tight all-but-many lossy trapdoor functions (ABM-LTFs) from LWE. These results are significant, as ABM-LTFs have several important applications in constructing other primitives, such as almost tight encryption schemes that are secure against selective opening attacks and CCA2 attacks (SO-CCA2) [17], and almost tight encryption schemes with multiple challenges against CCA2 attacks [41].

Despite these excellent advances, we however notice a common drawback in all prior almost tight lattice-based results – they all require super-polynomial moduli. It is much more favorable to build schemes with a polynomial modulus, as this provides a better security guarantee, e.g., a better approximate factor of worst-case lattice problems, and thus can lead to smaller parameters resulting in better efficiency. Additionally from a theoretic point of view, it is important to determine whether a super-polynomial modulus is inherent in achieving almost tight security in lattice-based crypto. Therefore, we ask:

Can we achieve (almost) tight security in lattices with a polynomial modulus ?

1.1 Our Results

In this work, we answer this question in a positive way for the following important primitives – PRF, IBE, and ABM-LTF. In particular, we construct and prove almost tight security of all these primitives with respect to LWE with polynomial moduli. Some other almost tight constructions can also be obtained along this line as we describe several examples. (1) Similar to the work of Boyen and Li [16], our technique of IBE can be used to derive almost tight signature schemes. Moreover, our IBE can be (almost) tightly extended to CCA2-IBE. (2) We can achieve almost tight IND-SO-CCA2 secure encryption schemes from LWE with a polynomial modulus q, following the framework of [17]. (3) We can achieve almost tight encryption schemes for multiple ciphertexts against CCA2 attacks from LWE with a polynomial modulus q, following the framework of [41]. Below we summarize our main results.

1. We prove that the GGM-based PRF in [8] is almost tight with respect to LWE with a polynomial modulus. This derives the *first* almost tight lattice-based PRF with a polynomial modulus. The crux relies on a new route of reduction LWE $\rightarrow$ Q-LWR$'$ $\rightarrow$ PRF, avoiding the known non-tight approach, i.e., LWE $\rightarrow$ PRG $\rightarrow$ PRF.[2]
   Moreover, our reduction LWE $\rightarrow$ Q-LWR$'$ has advantages over existing reductions: (1) we remove the additional number-theoretic limitation on the modulus in [4]; (2) our reduction has better running time and distinguishing probability than those in the work [11,16]. See Sects. 1.2 and 3 for further discussions.
2. We then construct an almost tight adaptively secure IBE from lattices with a polynomial modulus. This improves the prior work [16] by weakening its underlying assumption, i.e., LWE for some super-polynomial modulus. To achieve this, we first improve the framework of [16], showing that an almost tight PRF (even not computable in NC1) suffices for achieving almost tight IBE with a polynomial modulus. Then the desired IBE follows by combining our almost tight PRF (not necessarily in NC1) with the improved framework.
3. We further show that our technique in Contribution 2 can be used to achieve an almost tight ABM-LTF and signatures from LWE with a polynomial modulus, improving the underlying assumption needed in the prior work [17,41].

1.2 Our Techniques

Pseudorandom Functions

In this work, we derive the first almost tight PRF with respect to LWE with a polynomial modulus. To illustrate our new ideas, we first briefly review the elegant approach by Banerjee, Peikert, and Rosen [8], who constructed the first lattice-based PRF by introducing an intermediate problem – the learning

[2] Q is the number of queries in the PRF; LWR$'$ is a variant of the LWR problem originally defined in the work [8]; Q-LWR$'$ is a multi-secret variant of LWR$'$ that includes inner products of Q secrets per sample.

with rounding (LWR) assumption, a *de-randomized* version of the LWE assumption [49]. In LWR, there is a secret vector $\boldsymbol{s} \in \mathbb{Z}_q^n$ and the target is to distinguish $(\boldsymbol{a}, \lfloor\langle\boldsymbol{a}, \boldsymbol{s}\rangle\rceil_{q\to p})$ from the uniform distribution, where $(\boldsymbol{a}, \boldsymbol{s}) \xleftarrow{\$} \mathbb{Z}_q^n \times \mathbb{Z}_q^n$, and the rounding function is taken as $\lfloor x \mod q\rceil_{q\to p} = \lfloor x(p/q)\rceil \mod p$. Since then, the work [15] and follow-up work [7] have built PRFs based on the LWE/LWR (or their variants), and different reductions from LWE to LWR have been proved for various parameters [4,6,11].

We observe that all non-GGM PRFs [7,8] cannot be proved secure under LWE with a polynomial modulus using current techniques: (1) The synthesizer in Naor-Reigold-based PRFs [45] need to use LWR with unbounded samples. However, all known reductions from LWE to LWR [4,6,8,11] with polynomial moduli require that the number of samples is bounded; (2) Other constructions such as the direct construction [8], tree-based construction [7], and the key-homomorphic PRF [7,15,41], require the modulus to be larger than the noise, which grows super-polynomially as needed in their analyses.

On the other hand, the GGM-based PRFs can be proved secure under LWE with a polynomial modulus. This is because LWR with bounded samples suffices for the GGM analysis (see [4]), and we do know reductions from LWE to LWR with a polynomial modulus [4,6,11]. However, the reduction loss in this approach depends on the number of queries Q by the PRF adversary. This work shows how to remove this dependency on Q.

Our New Idea: A New Route of Reduction

We first recall that the GGM framework [31] showed that a length-doubling PRG implies a PRF. The proof of security can be decomposed into two steps (c.f. [37]), i.e., PRG $\xrightarrow{(1)}$ Q-PRG $\xrightarrow{(2)}$ PRF, where the Q-PRG problem is to distinguish Q independent samples of PRG from Q random strings. The second step is almost tight, yet the loss in the first step depends on Q under currently known hybrid proof techniques. Therefore, any route that starts with LWE → PRG will hit this technical difficulty. To bypass this barrier, we propose a new route:

$$\boxed{\mathsf{LWE} \xrightarrow{(i)} n\text{-}\mathsf{LWE} \xrightarrow{(ii)} Q\text{-}\mathsf{LWR}' \xrightarrow{(iii)} \mathsf{PRF},}$$

where the Q-LWR′ problem asks to distinguish samples either from $(\mathbf{A}, \lfloor\boldsymbol{s}_1^t \cdot \mathbf{A}\rceil_{q\to p}, \ldots, \lfloor\boldsymbol{s}_Q^t \cdot \mathbf{A}\rceil_{q\to p})$ or from the corresponding uniform distribution, where $\boldsymbol{s}_i \leftarrow \mathbb{Z}_p^n$ for $i \in [Q]$.[3]

The reduction loss in (i) is n by a simple hybrid argument, and thus almost tight. The reduction loss in (iii) is k (the input length), which is almost tight. It is worth pointing out that the n-LWE problem is also known as the multi-secret LWE problem. As n is a system parameter that only depends on the security parameter, sometimes this version of the LWE is used as the starting point of the underlying hard problem, e.g. the work [17].

[3] The original LWR problem [8] samples the secret uniformly at random from $\mathbb{Z}_q^n$.

We next present a new analysis of n-$\mathsf{LWE} \xrightarrow{(ii)} Q$-$\mathsf{LWR}'$, which can be proved *tight* (for some useful settings of parameters). To achieve this, we present a refinement of the work [4] below:

Refinement of [4]. We present a critical observation that the information-theoretic step of [4] can be applied to the multi-secret setting. More specifically, we take the steps as follows.

1. First, we break $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ into $(\bar{\mathbf{A}}, \boldsymbol{a}) \in \mathbb{Z}_q^{n\times(m-1)} \times \mathbb{Z}_q^n$ and switch $\bar{\mathbf{A}}$ into some lossy but indistinguishable $\tilde{\mathbf{A}}$. This incurs a security loss $\varepsilon_{n\text{-}\mathsf{LWE}}$.
2. Then, we prove that $(\tilde{\mathbf{A}}, \lfloor {\boldsymbol{s}_1}^t \cdot \tilde{\mathbf{A}} \rceil_{q\to p}, \cdots, \lfloor \boldsymbol{s}_Q^t \cdot \tilde{\mathbf{A}} \rceil_{q\to p}, \boldsymbol{a}, \lfloor \boldsymbol{a} \cdot \boldsymbol{s}_1 \rceil_{q\to p}, \cdots,$ $\lfloor \boldsymbol{a} \cdot \boldsymbol{s}_Q \rceil_{q\to p})$ is *statistically close* to $(\tilde{\mathbf{A}}, \lfloor \boldsymbol{s}_1^t \cdot \tilde{\mathbf{A}} \rceil_{q\to p}, \cdots, \lfloor \boldsymbol{s}_Q^t \cdot \tilde{\mathbf{A}} \rceil_{q\to p}, \boldsymbol{a},$ $\lfloor \boldsymbol{u}_1 \rceil_{q\to p}, \cdots, \lfloor \boldsymbol{u}_Q \rceil_{q\to p}))$ for truly random $\{\boldsymbol{u}_i\}_{i\in[Q]}$.
3. Next, we switch $\tilde{\mathbf{A}}$ back to $\bar{\mathbf{A}}$, with another security loss $\varepsilon_{n\text{-}\mathsf{LWE}}$.
4. Then we repeat the above steps for each column of $\mathbf{A}$.

The second step can be proved using the concept that a strong extractor extracts randomness from a block-source. It is clear that $(\boldsymbol{a}, \langle \boldsymbol{a}, \boldsymbol{s} \rangle)$ is a strong extractor. As we can show that $\boldsymbol{s}_1, \cdots, \boldsymbol{s}_Q$ form a block-source,[4] $\boldsymbol{a}$ can extract their randomness [52]. This step might incur a dependency on Q yet in the *purely information-theoretic* manner, i.e., the dependency on Q will not affect $\varepsilon_{n\text{-}\mathsf{LWE}}$ in the multiplicative way. With appropriate parameters, we can make the statistical distance in Step 2 arbitrarily small, e.g., 2^{-n}, and the security loss in Steps 1–3 would be $2\varepsilon_{n\text{-}\mathsf{LWE}} + 2^{-n}$. By repeating Steps 1–3 for all columns (i.e. m), we can obtain a reduction with loss $m(2\varepsilon_{n\text{-}\mathsf{LWE}} + 2^{-n})$, which is almost tight.

Further Improvements. Next, we present two optimizations of the above approach: (1) By using a more efficient hybrid analysis, we can get rid of the dependency on m in the above argument. Particularly, if the secret $\boldsymbol{s}$ has sufficient entropy relative to m, we can extract multiple columns per hybrid, resulting in using less hybrids and thus the overall reduction can be independent of m. (2) By using a leftover hash lemma for general modulus q with a more careful analysis, we can further remove the number-theoretic restrictions in [4]. This broadens the range of parameter selections – for example, the prior analysis [4] does not cover several useful settings, e.g., $q = p^e$, where our improvement does.

Putting Things Together for PRF. Putting things together, we are able to achieve: n-$\mathsf{LWE} \to \mathsf{PRF}$ with reduction loss k, and similarly $\mathsf{LWE} \to \mathsf{PRF}$ with reduction loss kn. By applying the technique of input-domain extension by [26], we can further reduce the loss k to $\omega(\log \kappa)$ and achieve the on-the-fly security. We summarize the results as follow:

Theorem 1.1 (Informal). *With some polynomial modulus q, we have: (1) n-$\mathsf{LWE} \to \mathsf{PRF}$ with reduction loss $\omega(\log \kappa)$, and (2) $\mathsf{LWE} \to \mathsf{PRF}$ with reduction loss $n \cdot \omega(\log \kappa)$.*

[4] More precisely, we can prove that $\boldsymbol{s}_1, \cdots, \boldsymbol{s}_Q$ have high min-entropy and form a block-source, conditioned on $\lfloor \boldsymbol{s}_1^t \cdot \tilde{\mathbf{A}} \rceil_{q\to p}, \cdots, \lfloor \boldsymbol{s}_Q^t \cdot \tilde{\mathbf{A}} \rceil_{q\to p}$.

A Note on Dimension Loss. For general moduli p, q, all known reductions $\mathsf{LWE} \to (Q\text{-})\mathsf{LWR}$ ([4,6,11] and ours) incur a dimension loss, i.e., LWE with dimension ℓ implies $(Q\text{-})\mathsf{LWR}$ with dimension ranging from $O(\ell)$ to $O(\ell \log q)$. As our almost tight result $\mathsf{LWE} \to Q\text{-}\mathsf{LWR}$ can achieve dimension loss of a constant factor, in the setting of general moduli, our reduction $\mathsf{LWE} \to \mathsf{PRF}$ is better than existing non-tight analyses $\mathsf{LWE} \to \mathsf{LWR} \to \mathsf{PRF}$ [4,6,11] in terms of security loss and in some cases as well dimension loss.

For special moduli p, q such that $p|q$, the reduction $\mathsf{LWE} \to \mathsf{LWR}$ of Bai et al. [6] does not incur a dimension loss, yet their reduction running time blows up significantly (at least quadratically) as the analysis goes through a decision to search step. An alternative approach would take the LWE function $f_{\mathbf{A}}(\boldsymbol{s}, \boldsymbol{e}) = \mathbf{A} \cdot \boldsymbol{s} + \boldsymbol{e}$ as a PRG, which is indeed length expanding as we do not need $n \log q$ bits of randomness to represent $\boldsymbol{e}$. This approach would not incur a dimension loss nor impose number theoretic restrictions on the modulus q. By using these two approaches, one can get a non-tight GGM PRF with the same dimension parameter as the underlying LWE, namely ℓ.

In general, a non-tight PRF (with dimension ℓ) and a tight PRF (with dimension $O(\ell)$) are incomparable as we discuss below. On one hand, if LWE is exponentially hard, e.g., $\varepsilon_{\mathsf{LWE}}(\ell) = 2^{-\ell}$, the non-tight PRF only needs to scale up ℓ to $(\ell + \log Q)$ to accommodate the security loss of a factor Q. In this case, the non-tight PRF parameter is better than the tight one. On the other hand, if LWE is only super-polynomially hard, e.g., $\varepsilon_{\mathsf{LWE}}(\ell) = 2^{-\log^2(\ell)}$, the non-tight PRF needs to scale up ℓ to $e\ell$ where $\log e \approx \log Q/(2 \log \ell)$, in order to accommodate the security loss. As e can be an arbitrary constant depending on the adversary, the tight PRF is better in this setting.

Almost Tight IBE and ABM-LTFs from LWE with Polynomial q
Recently, Boyen and Li [16] showed how to achieve an almost tight IBE from LWE by proposing a novel technique that applies (key) homomorphic evaluation on PRF. Shortly, this technique was used to achieve $\mathsf{ABM\text{-}LTF}$s from LWE and thus many of their applications [17,41]. However, their techniques inherently require a super-polynomial modulus in achieving almost tight security. Below, we present our new insights to remove this restriction. For simplicity of presentation, we just focus on the setting of IBE [16] and remark that the idea can be extended to the $\mathsf{ABM\text{-}LTF}$ in a similar way.

Basically, Boyen and Li [16] showed that an almost tight IBE can be constructed if (1) LWE is hard, (2) there exists an (almost) tight PRF that can be evaluated in NC1. Even though their reduction is tight from $\mathsf{LWE} + \mathsf{PRF}$, there is no known instantiation of the required PRF from LWE with a polynomial modulus. Therefore, there is no construction of pure lattice-based almost tight IBE with a polynomial modulus. How to achieve such a PRF instantiation is a natural and interesting open problem.

The GGM-PRF with our new analysis still does not solve the open problem directly, as the GGM-based construction is not known to be in NC1. Nevertheless, we bypass this issue by showing that the requirement on NC1 is not necessary. Particularly, we improve the framework of Boyen and Li [16] by showing that the following conditions are sufficient: (1) LWE is hard, (2) there exists an almost tight PRF, and (3) there exists a (leveled) fully homomorphic encryption scheme whose decryption algorithm can be computed in NC1.[5] Our desired IBE follows, as we can instantiate all the components from LWE with a polynomial modulus – the GGM-based PRF in this work for (2), and the FHE schemes [3,22] for (3). In summary, we achieve the following theorem:

Theorem 1.2 (Informal). *Assuming* LWE *is hard for some polynomial modulus q, there exists an almost tight adaptively secure* IBE *in the standard model.*

Below we highlight our new ideas. We first recall the framework of Boyen and Li [16], which can be described roughly as follows. The public key contains matrices $\mathbf{A}$ and $\mathbf{B}_1, \ldots, \mathbf{B}_k$. At various steps (in the proof), the matrices are encoded as $\mathbf{B}_i = \mathbf{A} \cdot \mathbf{R}_i + s_i \mathbf{G}$, where s_i is the i-th bit of a PRF key K and $\mathbf{R}_i$'s are random matrices with small norms. In the key derivation process, i.e., to derive $\mathsf{sk}_{\mathsf{id}}$, their scheme applies the (key) homomorphic evaluation algorithm [14] on the matrices $\{\mathbf{B}_i\}_{i \in k}$ to compute the function $\mathsf{PRF}(K, \mathsf{id})$ for some given id, resulting in $\mathbf{B}_{\mathsf{id}} = \mathbf{A} \cdot \mathbf{R}_{\mathsf{id}} + \mathsf{PRF}(K, \mathsf{id})\mathbf{G}$. Their IBE scheme [16] requires that $\|\mathbf{R}_{\mathsf{id}}\| < q$, as $\|\mathbf{R}_{\mathsf{id}}\|$ affects the quality of the SampleRight algorithm and the noise growth. As long as the PRF computation is in NC1 [16], then $\|\mathbf{R}_{\mathsf{id}}\|$ can be upper bounded by a polynomial, allowing the scheme to use a polynomial modulus q. On the other hand, if the PRF is not computable in NC1, then a super-polynomial q seems to be inherent in this approach as $\|\mathbf{R}_{\mathsf{id}}\|$ would become super-polynomial.

To bypass the technical barrier, we introduce a two-step approach that integrates homomorphic evaluation on leveled HE ciphertexts, key homomorphic evaluation on the public matrices, and Gentry's bootstrapping technique [3,29]. Given a leveled FHE (HE) that supports homomorphic computation of the PRF and has an NC1 decryption algorithm, we add an encryption of a PRF key K, i.e., $c \leftarrow \mathsf{HE.Enc}(K)$, to the public key, and encode $\mathbf{B}_i = \mathbf{A} \cdot \mathbf{R}_i + (\mathsf{sk})_i \mathbf{G}$, where $(\mathsf{sk})_i$ is the i-th bit of the decryption key of the HE scheme. Then our new key derivation process consists of the following two steps:

1. (Homomorphic Evaluation of PRF) First run $\tilde{c} = \mathsf{HE.Eval}(\mathsf{PRF}(\cdot, \mathsf{id}), c)$ to homomorphically evaluate $\mathsf{PRF}(K, \mathsf{id})$.
2. (Key Homomorphic Bootstrapping) Next run the key homomorphic evaluation of the decryption algorithm of HE on the matrices $\{\mathbf{B}_i\}_{i \in [k]}$ with the input $\tilde{c}$, i.e., evaluate $\mathsf{HE.Dec}(\mathsf{sk}, \tilde{c})$ homomorphically. Then we obtain $\mathbf{B}_{\mathsf{id}} = \mathbf{A} \cdot \mathbf{R}_{\mathsf{Dec}} + \mathsf{PRF}(K, \mathsf{id})\mathbf{G}$.

As the decryption algorithm can be computed in NC1, we know that $\|\mathbf{R}_{\mathsf{Dec}}\|$ can be bounded by a polynomial. Furthermore, we know that the required HE

[5] Actually a homomorphic encryption that supports evaluation of the PRF in (2) suffices.

can be instantiated from LWE with a polynomial modulus [3,22]. Putting all things together, we can obtain the desired IBE.

We note that our result above *does not* need the circular security assumption, as we only need a leveled HE that supports computation of the PRF, which is of a bounded depth. Moreover, in our key homomorphic bootstrapping step, the secret key of HE is information-theoretically hidden in the matrices $\mathbf{B}_i$'s. This again does not rely on the circular security assumption.

Finally, we observe that the above two-step approach can be used to improve the modulus used in prior $\mathsf{ABM\text{-}LTF}$ [17,41] and signatures [16]. Particularly, we achieve:

Theorem 1.3 (Informal). *Assuming* LWE *is hard for some* $q = \mathsf{poly}(\kappa)$*, there exist an almost tight* $\mathsf{ABM\text{-}LTF}$ *and a signature scheme with a poly modulus.*

Other Related Work. Very recently, Jager *et al.* [34] proposed a new framework to improve the size of secret key and reduction loss of the PRFs [8,40,45], yet their instantiations from lattices however, still require super-polynomial moduli.

2 Preliminaries

Notations. We let κ denote the security parameter. For an integer n, let $[n]$ denote the set $\{1, ..., n\}$. We use bold lowercase letters (e.g. $\boldsymbol{a}$) to denote vectors and bold capital letters (e.g. $\mathbf{A}$) to denote matrices. For a positive integer $q \geq 2$, let $\mathbb{Z}_q$ be the ring of integers modulo q. For a distribution or a set X, we write $x \overset{\$}{\leftarrow} X$ to denote the operation of sampling an uniformly random x according to X. For distribution X, Y, we let $\mathsf{SD}(X, Y)$ denote their statistical distance. We write $X \overset{s}{\approx} Y$ to mean that they are statistically close, and $X \overset{c}{\approx} Y$ to say that they are computationally indistinguishable. We let $\mathsf{negl}(\kappa)$ denote the set of all negligible function $\mu(\kappa) = \kappa^{-\omega(1)}$.

Definition 2.1 (Computational indistinguishability). *We say that two experiments* H_0, H_1 *are* (t, ε)*-indistinguishable with oracle access if for every distinguisher* $\mathcal{D}$ *within running time* t*, we have* $|\mathsf{Pr}[\mathcal{D}^{H_0}\mathsf{accepts}] - \mathsf{Pr}[\mathcal{D}^{H_1}\mathsf{accepts}]| < \varepsilon$*, where the probabilities are taken over the coin tosses of* H_0, H_1.

2.1 Learning with Error

We define the multi-secret variant of learning with error, i.e., N-LWE, and note that the standard learning with error can be denoted as 1-LWE.

Definition 2.2 (Multi-secret Learning with Errors (LWE) Assumption [49]). *Let* κ *be the security parameter,* n, m, q, N *be integers (functions of* κ*), and* $\chi = \chi(\kappa)$ *be a distribution over* $\mathbb{Z}_q$*. The* N-$\mathsf{LWE}_{n,m,q,\chi}$ *assumption with parameter* N *can be stated that for independently sampled* $\mathbf{A} \overset{\$}{\leftarrow} \mathbb{Z}_q^{n \times m}$*,* $\boldsymbol{u}_i \overset{\$}{\leftarrow} \mathbb{Z}_q^m$*,*

$\boldsymbol{s}_i \xleftarrow{\$} \mathbb{Z}_q^n$ and $\boldsymbol{e}_i \xleftarrow{\$} \chi^m$ for $i \in [N]$, the following distributions are computationally indistinguishable: $(\mathbf{A}, (\boldsymbol{s}_1^t \cdot \mathbf{A} + \boldsymbol{e}_1^t), \ldots, (\boldsymbol{s}_N^t \cdot \mathbf{A} + \boldsymbol{e}_N^t)) \overset{c}{\approx} (\mathbf{A}, \boldsymbol{u}_1^t, \ldots, \boldsymbol{u}_N^t)$. We say N-$\mathsf{LWE}_{n,m,q,\chi}$ problem is (t, ε)-hard if the two distributions above are (t, ε)-indistinguishable.

By a simple hybrid argument, we can derive a reduction from 1-$\mathsf{LWE}_{n,m,q}$ to N-$\mathsf{LWE}_{n,m,q}$ with a security loss with a multiplicative factor of N. The work [20, 47, 49] showed that there exist quantum/classical reductions from some worst-case lattice problems ($\mathsf{GapSVP}, \mathsf{SIVP}$) to the LWE problem.

2.2 Learning with Rounding

For any integer modulus $q > 2$, $\mathbb{Z}_q$ denotes the quotient ring of integers modulus q. We define a rounding function $\lfloor \cdot \rceil_p : \mathbb{Z}_q \to \mathbb{Z}_p$ for $q \geq p \geq 2$ as

$$\lfloor x \rceil_{q \to p} = \lfloor (p/q)\bar{x} \rceil_{q \to p},$$

where $\bar{x} \in \mathbb{Z}$ is any integer congruent to x mod q. Furthermore, $\lfloor \cdot \rceil_{q \to p}$ can be extended component-wise to vectors and matrices over $\mathbb{Z}_q$. In places where the context is clear about the modulus q, we would omit q in the notation as $\lfloor \cdot \rceil_p$ for simplicity of presentation.

Similar to the multi-secret LWE, we define a multi-secret variant for the LWR assumption, and note that the original LWR [8] can be denoted as 1-LWR.

Definition 2.3 (Multi-secret LWR). *Let $\kappa \geq 1$ be the security parameter, $n, q \geq p \geq 2$, Q be integers (functions of κ). The Q-$\mathsf{LWR}_{n,m,q,p}$ assumption states that for independently sampled $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, $\boldsymbol{u}_i \xleftarrow{\$} \mathbb{Z}_q^m$, $\boldsymbol{s}_i \xleftarrow{\$} \mathbb{Z}_q^n$ with $i \in [Q]$, the following distributions are computationally indistinguishable:*

$$(\mathbf{A}, \lfloor \boldsymbol{s}_1^t \cdot \mathbf{A} \rceil_p, \ldots, \lfloor \boldsymbol{s}_Q^t \cdot \mathbf{A} \rceil_p) \overset{c}{\approx} (\mathbf{A}, \lfloor \boldsymbol{u}_1^t \rceil_p, \ldots, \lfloor \boldsymbol{u}_Q^t \rceil_p),$$

We say the Q-$\mathsf{LWR}_{n,m,q,p}$ problem is (t, ε)-hard if the two distributions above are (t, ε)-indistinguishable.

Below we define a variant of the LWR problem, namely, LWR′, which will be useful for our PRF construction.

Definition 2.4 (Multi-secret LWR′). *The Q-$\mathsf{LWR}'_{n,m,q,p}$ problem is the same as Q-$\mathsf{LWR}_{n,m,q,p}$ except that the secret vectors $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_Q$ are sampled from $\mathbb{Z}_p^n$.*

2.3 Pseudorandom Function and Identity-Based Encryption

Definition 2.5 (Pseudorandom function). *Let A and B be finite sets, and let $\mathcal{F} = \{F_i : A \to B\}$ be a function family, endowed with efficient sampleable distribution ($\mathcal{F}, A$ and B are all indexed by the security parameter λ). We say that $\mathcal{F}$ is a (t, Q, ε)-pseudorandom function(PRF) family if the following two experiments are (t, ε)-indistinguishable with oracle access up to Q adaptive queries: (1) Choose a function $F \leftarrow \mathcal{F}$, and (2) Choose a uniformly random function $R : A \to B$.*

Definition 2.6 (Identity-Based Encryption (IBE) [13,51]). *An identity-based encryption scheme consists of four* PPT *algorithms* $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *defined as follows:*

- $\mathsf{Setup}(1^\kappa)$*: Given the security parameter, it outputs a master public key* mpk *and a master secret key* msk*.*
- $\mathsf{KeyGen}(\mathsf{msk}, \mathsf{id})$*: Given the* msk *and an identity* $\mathsf{id} \in \{0,1\}^\ell$*, it outputs the identity secret key* $\mathsf{sk}_{\mathsf{id}}$*.*
- $\mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, m)$*: Given the* mpk*, an identity* $\mathsf{id} \in \{0,1\}^\ell$*, and a message* m*, it outputs a ciphertext* c*.*
- $\mathsf{Dec}(\mathsf{sk}_{\mathsf{id}}, c)$*: Given a secret key* $\mathsf{sk}_{\mathsf{id}}$ *for identity* id *and a ciphertext* c*, it outputs a plaintext* m*.*

The following correctness and security properties must be satisfied:

Correctness: For all security parameter κ, identity $\mathsf{id} \in \{0,1\}^\ell$ and message m, the following holds: $\Pr[\mathsf{Dec}(\mathsf{sk}_{\mathsf{id}}, \mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, m)) \neq m] = \mathsf{negl}(\kappa)$, where $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{id})$ and $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\kappa)$.

Security: We define the *adaptive* chosen-plaintext security (IND-ID-CPA) for IBE as below, where the adversary can adaptively make secret key queries.

Experiment $(\mathsf{IND\text{-}ID\text{-}CPA}^{\mathsf{IBE}}(\mathcal{A}))$

1. $(\mathsf{mpk}, \mathsf{msk}) \xleftarrow{\$} \mathsf{Setup}(1^\kappa)$.
2. $(\mathsf{id}^*, m_0, m_1) \xleftarrow{\$} \mathcal{A}_1^{\mathsf{KeyGen}(\mathsf{msk},\cdot)}(\mathsf{mpk})$ where $|m_0| = |m_1|$ and for each query id by $\mathcal{A}_1$ to $\mathsf{KeyGen}(\mathsf{msk},\cdot)$ we have that $\mathsf{id} \neq \mathsf{id}^*$.
3. $b \xleftarrow{\$} \{0,1\}$.
4. $m^* = m_b$
5. $c^* \xleftarrow{\$} \mathsf{Enc}(\mathsf{mpk}, \mathsf{id}^*, m^*)$
6. $b' \xleftarrow{\$} \mathcal{A}_2^{\mathsf{KeyGen}(\mathsf{msk},\cdot)}(\mathsf{mpk}, c^*)$ where for each query id by $\mathcal{A}_2$ to $\mathsf{KeyGen}(\mathsf{msk},\cdot)$ we have that $\mathsf{id} \neq \mathsf{id}^*$.
7. Output 1 if $b^* = b'$ and 0 otherwise.

Definition 2.7. *For a security parameter* κ*, let* $t = t(\kappa), q = q(\kappa)$ *and* $\varepsilon = \varepsilon(\kappa)$*. we say that an* IBE *scheme* $\mathcal{E}$ *is* (t, q, ε)-IND-ID-CPA *secure if for any* t *time adversary* $\mathcal{A}$ *makes at most* q *secret key queries and the following holds:*

$$\Pr[\mathsf{IND\text{-}ID\text{-}CPA}^{\mathsf{IBE}}(\mathcal{A}) = 1] \leq \frac{1}{2} + \varepsilon(\kappa).$$

2.4 Lattice Backgrounds

Theorem 2.8 (Trapdoor Generation [2,43]). *There is a probabilistic polynomial-time algorithm* $\mathsf{TrapGen}(1^n, q, m)$ *that for all* $m \geq m_0 = m_0(n, q) = O(n \log q)$*, outputs* $(\mathbf{A}, \mathbf{T_A})$ *s.t.* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *is within statistical distance* 2^{-n} *from uniform and the distribution of* $\mathbf{T_A}$ *is the Discrete Gaussian* $D_{Z^m,\tau}$ *conditioned on* $\mathbf{A} \cdot \mathbf{T_A} = 0 \pmod q$ *and* $\tau = O\sqrt{n \log q \log n}$*.*

Theorem 2.9 **([1]).** *Let $q > 2, m > n$. (i) If $s > \|\tilde{\mathbf{T}}_{\mathbf{A}}\| \cdot \omega(\sqrt{\log(m+m_1)})$. Then there exists an algorithm* $\mathsf{SampleLeft}$ *taking $(\mathbf{A} \in \mathbb{Z}_q^{n\times m}, \mathbf{B} \in \mathbb{Z}^{n\times m_1}, \mathbf{T}_{\mathbf{A}}, \boldsymbol{u} \in \mathbb{Z}_q^n, s)$ as input, outputs a vector $\boldsymbol{d} \in \mathbb{Z}^{m+m_1}$ distributed statistically close to $D_{\Lambda_q^{\boldsymbol{u}}([\mathbf{A}|\mathbf{B}]),s}$. (ii) If $s > \|\tilde{\mathbf{T}}_{\mathbf{B}}\| \cdot \|\mathbf{R}\| \cdot \omega(\sqrt{\log m})$. Then there exists an algorithm* $\mathsf{SampleRight}$ *taking $(\mathbf{A} \in \mathbb{Z}_q^{n\times k}, \mathbf{R} \in \mathbb{Z}^{k\times m}, \mathbf{B} \in \mathbb{Z}^{n\times m}, \mathbf{T}_{\mathbf{B}}, \boldsymbol{u} \in \mathbb{Z}_q^n, s)$ as input, outputs a vector $\boldsymbol{d} \in \mathbb{Z}^{m+k}$ distributed statistically close to $D_{\Lambda_q^{\boldsymbol{u}}([\mathbf{A}|\mathbf{AR}+\mathbf{B}]),s}$.*

Gadget Matrix. We recall the "gadget matrix" $\mathbf{G}$ defined in [43]. The "gadget matrix" $\mathbf{G} = \boldsymbol{g} \otimes \mathbf{I}_n \in \mathbb{Z}_q^{n\times n\lceil \log q\rceil}$ where $\boldsymbol{g} = (1, 2, 4, ..., 2^{\lceil \log q\rceil - 1})$.

Lemma 2.10 **([43], Theorem 1).** *Let q be a prime, and n, m be integers with $m = n\lceil \log q\rceil$. There is a full-rank matrix $\mathbf{G} \in \mathbb{Z}_q^{n\times m}$ such that the lattice $\Lambda_q^{\perp}(\mathbf{G})$ has a publicly known trapdoor matrix $\mathbf{T}_{\mathbf{G}} \in \mathbb{Z}^{n\times m}$ with $\|\tilde{\mathbf{T}}_{\mathbf{G}}\| \le \sqrt{5}$, where $\tilde{\mathbf{T}}_{\mathbf{G}}$ is the Gram-Schmidt order orthogonalization of $\mathbf{T}_{\mathbf{G}}$.*

Lemma 2.11 **([14], Lemma 2.1).** *There is a deterministic algorithm, denoted by $\mathbf{G}^{-1}(\cdot) : \mathbb{Z}_q^{n\times m} \to \mathbb{Z}^{m\times m}$, that takes any matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ as input, and outputs the preimage $\mathbf{G}^{-1}(\mathbf{A})$ of $\mathbf{A}$ such that $\mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{A}) = \mathbf{A} \pmod q$ and $\|\mathbf{G}^{-1}(\mathbf{A})\| \le m$.*

Definition 2.12 **($\boldsymbol{\delta}$-compatible algorithms [54]).** *We say that the deterministic algorithms $(\mathsf{Eval}^{\mathsf{Pub}}, \mathsf{Eval}^{\mathsf{Trap}})$ are δ-compatible for a function family $\mathcal{F} = \{f : \{0,1\}^{\ell} \to \{0,1\}\}$ if they are efficient and satisfy the following properties:*

- $\mathsf{Eval}^{\mathsf{Pub}}(f \in \mathcal{F}, \{\mathbf{A}_i \in \mathbb{Z}_q^{n\times m}\}_{i\in[\ell]}) = \mathbf{A}_f \in \mathbb{Z}^{n\times m}$.
- $\mathsf{Eval}^{\mathsf{Trap}}(f \in \mathcal{F}, \mathbf{A}, \boldsymbol{x} \in \{0,1\}^{\ell}, \{\mathbf{R}_i \in \mathbb{Z}^{m\times m}\}_{i\in[\ell]}) = \mathbf{R}_f \in \mathbb{Z}^{m\times m}$.

For any $\boldsymbol{x} = (x_1, ..., x_{\ell}) \in \{0,1\}^{\ell}$, we require that the following holds:

$$\mathsf{Eval}^{\mathsf{Pub}}(f, \{\mathbf{AR}_i + x_i\mathbf{G}\}_{i\in[\ell]}) = \mathbf{AR}_f + f(\boldsymbol{x})\mathbf{G} \pmod q,$$

and we have $\|\mathbf{R}_f\|_{\infty} \le \delta \cdot \max_{i\in[\ell]}\{\|\mathbf{R}_i\|\}$.

Lemma 2.13 **(Noise Rerandomization [36]).** *Let q, ℓ, m be positive integers and r a positive real satisfying $r > max\{\eta_{\epsilon}(\mathbb{Z}^m), \eta_{\epsilon}(\mathbb{Z}^{\ell})\}$. Let $\boldsymbol{b} \in \mathbb{Z}_q^m$ be arbitrary vector and $\boldsymbol{x}$ chosen from $D_{\mathbb{Z}^m,r}$. Then for any $\mathbf{V} \in \mathbb{Z}^{m\times\ell}$ and positive real $\sigma > \mathsf{s}_1(\mathbf{V})$, there exists a PPT algorithm $\mathsf{ReRand}(\mathbf{V}, \boldsymbol{b}+\boldsymbol{x}, r, \sigma)$ that outputs $\boldsymbol{b}^{'} = \boldsymbol{b}\mathbf{V} + \boldsymbol{x}^{'} \in \mathbb{Z}_q^{\ell}$ where the statistical distance of the discrete Gaussian $D_{\mathbb{Z}^{\ell},2r\sigma}$ and the distribution of $\boldsymbol{x}^{'}$ is within 8ϵ.*

Fully Homomorphic Encryption. We present the syntax of (leveled fully) homomorphic encryption. A homomorphic encryption scheme $\mathsf{HE} = (\mathsf{HE.KeyGen}, \mathsf{HE.Enc}, \mathsf{HE.Dec}, \mathsf{HE.Eval})$ is a quadruple of PPT algorithms as follows:

- $\mathsf{HE.KeyGen}(1^{\kappa})$. Generate an encryption key ek. a public evaluation key evk, and a secret decryption key dk.
- $\mathsf{HE.Enc}(\mathsf{ek}, \mu)$. Generate a ciphertext ct.
- $\mathsf{HE.Dec}(\mathsf{dk}, \mathsf{ct})$. Decrypt the ciphertext and output message μ.

- $\mathsf{HE.Eval}(\mathsf{evk}, f, \{\mathsf{ct}_i\})$. The algorithm takes evk and a function (circuit) f and a set of ciphertexts $\{\mathsf{ct}_i\}$ as input, and outputs an evaluated ciphertext ct_f.

Correctness and security follow by the standard definitions as [21,29]. If a homomorphic scheme HE supports evaluation of a class of functions $\mathcal{C}$, then it is $\mathcal{C}$-homomorphic. A fully homomorphic encryption supports evaluation of all polynomial-sized circuits. Details are deferred to full version of this paper.

Next, we present an important result, saying that for most of the LWE-based FHEs, the decryption circuits are in NC1 and can be homomorphically evaluated with a small noise growth.

Theorem 2.14 ([3,22]). *For all $n, q, m, \ell \in \mathbb{N}$, and for any sequence of matrices $(\mathbf{B}_1, ..., \mathbf{B}_\ell) \in (\mathbb{Z}_q^{n\times m})^\ell$ where $\mathbf{B}_i = \mathbf{A}\mathbf{R}_i + x_i\mathbf{G}$ for $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}, \mathbf{R}_i \xleftarrow{\$} \{-1,1\}^{m\times m}, x_i \xleftarrow{\$} \{0,1\}$, the following holds. For the special decryption algorithms $f \in \{0,1\}^\ell \to \{0,1\}$ of* LWE *based FHE [3,22], $\mathsf{Eval}^{\mathsf{Pub}}(f, \mathbf{B}_1, ..., \mathbf{B}_\ell) = \mathbf{A}\mathbf{R}_f + f(\boldsymbol{x})\mathbf{G} \pmod q$, where $\boldsymbol{x} = (x_1, ..., x_\ell)$, and $\|\mathbf{R}_f\|_2 \le O(n^{2+\varepsilon})$ for any $\varepsilon \in (0,1)$. In other word, the algorithms $(\mathsf{Eval}^{\mathsf{Pub}}, \mathsf{Eval}^{\mathsf{Trap}})$ are $O(n^{2+\varepsilon})$-compatible in this case.*

3 Almost Tight Lattice-Based PRF Under Poly Moduli

In this section, we first present an (almost) tight reduction of $\mathsf{LWE} \to Q\text{-}\mathsf{LWR}'$ for bounded number of samples with a polynomial modulus. This new reduction serves as the core technique to prove the almost tight security of GGM PRF from LWE with polynomial modulus.

3.1 LWR with a General Modulus q

To study the LWR problem with a general modulus q, we first present a useful leftover hash lemma in a general $\mathbb{Z}_q$. In particular, we show that matrix multiplication in general $\mathbb{Z}_q$ is a good extractor, i.e. $(\mathbf{A}, \boldsymbol{s}^t\mathbf{A}) \overset{s}{\approx} (\mathbf{A}, \boldsymbol{u})$, as long as the min-entropy of $\boldsymbol{s}$ mod p' has sufficient entropy for every factor p' of q.

We note that this condition for entropy is necessary as otherwise, we can construct a simple counterexample where the output distribution of $\boldsymbol{s}^t\mathbf{A}$ is far from uniform. Consider $q = 2^{10}$, and $\boldsymbol{s}$ is sampled uniformly from $\{0,2\}^n$. It is clear that $\boldsymbol{s}$ has min-entropy n and all components of $\boldsymbol{s}$ are small, but for any vector $\boldsymbol{a} \in \mathbb{Z}_q^n$, $\langle \boldsymbol{s}, \boldsymbol{a}\rangle$ is an even number and thus the distribution of $\langle \boldsymbol{s}, \boldsymbol{a}\rangle$ over a random $\boldsymbol{a}$ is far from uniform over $\mathbb{Z}_q$.

More formally, we use the following lemma to show that this entropy condition is sufficient for extraction.

Theorem 3.1 (Randomness Extraction for General q). *Let $z, n, k, q \in \mathbb{N}$ be integers and $\varepsilon \in (0,1)$ such that*

$$k > z\log q + 3(\log(zq) + \log(1/\varepsilon)) + 2(\log q)(\log\log q) + 7.$$

Suppose $\boldsymbol{s}$ is chosen from some distribution over $\mathbb{Z}_q^n$ such that $H_\infty(\boldsymbol{s} \bmod p) \geq k$ for any factor p of q, and $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times z}$, $\boldsymbol{u} \xleftarrow{\$} \mathbb{Z}_q^z$ are chosen independently of $\boldsymbol{s}$ from the uniform distribution. Then we have: $\Delta[(\mathbf{A}, \boldsymbol{s}^t \cdot \mathbf{A}); (\mathbf{A}, \boldsymbol{u}^t)] \leq \varepsilon$.

This theorem can be proved via Lemma 2.3 in [42]. We describe our alternative proof for completeness of presentation in the full version of this paper.

Next, we define a generalization of the weak learning with rounding (wLWR) assumption (in the form of multi-secret) in general $\mathbb{Z}_q$. Intuitively, the wLWR problem considers scenarios where the secret $\boldsymbol{s}$ comes from some high min-entropy distribution (e.g., perhaps the secret is somewhat leaked) instead of the uniform distribution.[6]

Definition 3.2 (Multi-secret wLWR). *Let κ be the security parameter, $n, m, q \geq p \geq 2, \gamma, k, Q$ be integers (functions of κ). The Q-$\mathsf{wLWR}^{(\gamma,k)}_{n,m,q,p}$ assumption states: let $\{(\boldsymbol{s}_i, \mathsf{aux}_i)\}_{i\in[Q]}$ be Q pairs of correlated random variables where (i) each pair is sampled independently of the others, (ii) the support of each $\boldsymbol{s}_i \in [-\gamma, \gamma]^n$, and (iii) $H_\infty(\boldsymbol{s}_i \bmod p' \mid \mathsf{aux}_i) \geq k$ for every prime factor p' of q and for $i \in [Q]$. Then the distributions below are computationally indistinguishable:*

$$(\{\mathsf{aux}_i\}_{i\in[Q]}, \mathbf{A}, \lfloor \boldsymbol{s}_1^t \cdot \mathbf{A}\rceil_p, \ldots, \lfloor \boldsymbol{s}_Q^t \cdot \mathbf{A}\rceil_p) \overset{c}{\approx} (\{\mathsf{aux}_i\}_{i\in[Q]}, \mathbf{A}, \lfloor \boldsymbol{u}_1\rceil_p, \ldots, \lfloor \boldsymbol{u}_Q\rceil_p),$$

where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}, \boldsymbol{u}_1, \cdots, \boldsymbol{u}_Q \xleftarrow{\$} \mathbb{Z}_q^m$ are chosen randomly and independently of $\{(\boldsymbol{s}_i, \mathsf{aux}_i)\}_{i\in[Q]}$. We say the Q-$w\mathsf{LWR}^{(\gamma,k)}_{n,m,q,p}$ problem is (t, ε)-hard if the two distributions above are (t, ε)-indistinguishable.

We remark that contrast with the previous definition by [4] for restricted moduli, our generalized definition instead impose more condition on the secret distribution, just as required in the randomness extraction in Theorem 3.1, i.e., $\boldsymbol{s} \bmod p'$ has sufficient entropy for every factor p' of q. Intuitively, without this additional condition in general $\mathbb{Z}_q$, $\lfloor \boldsymbol{s}^t \cdot \mathbf{A}\rceil$ might be far from uniform for some $\boldsymbol{s}$ which is only guaranteed to have high min-entropy.

More formally, we establish the following main theorem to show that Q-wLWR is at least as hard as n-LWE for a wide range of parameters.

Theorem 3.3 (Hardness of Multi-secret (w)LWR). *Let $k, \ell, n, m, p, q, \gamma, Q, \lambda$ be positive integers, $p_{\min}$ be the smallest prime factor of q, c be an integer, and χ be a β-bounded distribution for some real $\beta > 0$, such that $q \geq 2\beta\gamma nmp$. Assume n-$\mathsf{LWE}_{\ell,m,q,\chi}$ problem is (t, ε)-hard, then we have the following:*

- *(High entropy secret). Q-$\mathsf{wLWR}^{(\gamma,k)}_{n,m,q,p}$ is (t', ε')-hard, where $t' = t - \mathsf{poly}(\kappa), \varepsilon' = 2c\varepsilon + (Qc + 1)\frac{1}{2^\lambda}$, if $k \geq \big(\lfloor \frac{m}{c} \rfloor + 2(\log\log q) + \ell + \lambda + 3\big) \log q + 3 \log\lfloor \frac{m}{c} \rfloor + 3\lambda + 7$.*

[6] In prior work [4], the wLWR problem is originally defined with respect to the secret $\boldsymbol{s}$ having sufficient min-entropy, and it is proved hard just for restricted moduli q.

- *(Uniform secret).* Q-$\mathsf{LWR}_{n,m,q,p}$ *is* (t',ε')*-hard, where* $t' = t - \mathsf{poly}(\kappa), \varepsilon' = 2c\varepsilon + (Qc+1)\frac{1}{2^\lambda}$*, if* $n \geq \frac{1}{\min\{\log(2\gamma), \log(p_{\min})\}}\left(\left(\lfloor\frac{m}{c}\rfloor + 2(\log\log q) + \ell + \lambda + 3\right)\log q + 3\log\lfloor\frac{m}{c}\rfloor + 3\lambda + 7\right)$.

The proof of this theorem relies on the use of a lossy matrix and randomness extraction alternately as we described in Sect. 1.2. Due to space limit, we defer the full proof to the supplementary material in full version of this paper.

Note that the reduction loss in Theorem 3.3 does not depend on Q in the multiplicative way, and thus can be made tight in several parameter settings. Furthermore, the hardness of ordinary wLWR, LWR and LWR' in the general $\mathbb{Z}_q$ can be derived easily from this theorem.

As we discussed in the beginning of this section, our result in Theorem 3.3 improves the prior work [4] in the following two aspects: (1) our q does not require the additional number theoretic requirement, and (2) if the secret $\boldsymbol{s}$ has sufficient entropy, we can further improve the security loss. The work [4] can be thought as $c = m$ in our case.

Using the above theorem, we can prove the problem $\mathsf{LWR}'_{n,m,q,p}$ as a special case of the problem $\mathsf{wLWR}^{(\gamma,k)}_{n,m,q,p}$, where $\gamma = p$, and $k = n\left(\min\{\log p, \log(p_{\min})\}\right)$.

We note that by a simple calculation, $\boldsymbol{s} \xleftarrow{\$} \mathbb{Z}_p$ implies $H_\infty(\boldsymbol{s} \bmod p') \geq n\left(\min\{\log p, \log(p_{\min})\}\right)$ for any prime factor p' of q. Thus we have the following corollary.

Corollary 3.4 (Hardness of Multi-secret LWR'). *Let* $\ell, n, m, p, q, Q, \lambda$ *be positive integers,* $p_{\min}$ *be the smallest prime factor of* q*,* c *be an integer, and* χ *be a* β*-bounded distribution for some real* $\beta > 0$*, such that* $q \geq 2\beta nmp^2$*. Assume* n-$\mathsf{LWE}_{\ell,m,q,\chi}$ *problem is* (t,ε)*-hard, then* Q-$\mathsf{LWR}'_{n,m,q,p}$ *is* (t',ε')*-hard, where* $t' = t - \mathsf{poly}(\kappa), \varepsilon' = 2c\varepsilon + (Qc+1)\frac{1}{2^\lambda}$*, if* $n \geq \frac{1}{\min\{\log p, \log(p_{\min})\}}\left(\left(\lfloor\frac{m}{c}\rfloor + 2(\log\log q) + \ell + \lambda + 3\right)\log q + 3\log\lfloor\frac{m}{c}\rfloor + 3\lambda + 7\right)$.

Some Useful Setting of Parameters. Our reduction of $\mathsf{LWE} \rightarrow Q$-$\mathsf{LWR}'$ holds for a wide range of parameters (e.g., $q = p^e$). Here we describe one example, which will be used in our almost tight PRF in Sect. 3.2.

Table 1. Simple example of parameter setting

Parameters	Description	Setting
κ	Security parameter	
n	LWR dimension	50κ
m	Number of LWR samples	$2n$
p	Modulus of LWR	κ
q	Modulus of LWE	p^6
ℓ	LWE dimension	κ
c	Reduction parameter	24
λ	Statistical loss parameter	2κ
β	LWE error bound	$\sqrt{\kappa}$

Through combining Theorem 3.3 and Corollary 3.4, together with the parameter setting in Table 1, we can directly achieve the following corollary

Corollary 3.5. *Let κ be the security parameter, $\ell, n, m, p, q, \lambda, \beta, c$ be function of κ setting above. Assume n-$\mathsf{LWE}_{\ell,m,q,\chi}$ problem is (t, ε)-hard, then Q-$\mathsf{LWR}'_{n,m,q,p}$ is (t', ε')-hard for any $Q = \mathsf{poly}(\kappa)$ and sufficient large κ, where $t' = t - \mathsf{poly}(\kappa), \varepsilon' \leq 48\varepsilon + \frac{24Q+1}{2^{2\kappa}}$.*

3.2 Lattice-Based PRF with poly Modulus

In this section, we show that the GGM-based construction [8], when instantiated under LWR' with parameters as Table 1, indeed achieves almost tight security. Thus, we achieve the first almost tight LWE-based PRF with a poly modulus.

Lattice PRF via GGM. By using the (n)-LWR' (with bounded samples) and the GGM construction, one can derive a PRF, as shown by the work [8]. For completeness, below we include the construction, parameters, and a theorem that summarizes security.

Construction. For parameters $n \in \mathbb{N}$, moduli $q \geq p \geq 2$, and input length $k \geq 1$, the family $\mathcal{F}$ consists of functions from $\{0,1\}^k$ to $\mathbb{Z}_p^n$. A function $F \in \mathcal{F}$ is indexed by some $\mathbf{A}_0, \mathbf{A}_1 \in \mathbb{Z}_q^{n \times n}$ and $\boldsymbol{s} \in \mathbb{Z}_p^n$, and is defined as

$$F(x) = F_{\boldsymbol{s},\{\mathbf{A}_i\}_{i\in\{0,1\}}}(x_1, ..., x_k) ;= \lfloor \ldots \lfloor \lfloor \boldsymbol{s}^t \cdot \mathbf{A}_{x_1} \rceil_p \cdot \mathbf{A}_{x_2} \rceil_p \ldots \cdot \mathbf{A}_{x_k} \rceil_p .$$

We endow $\mathcal{F}$ with the distribution where $\{\mathbf{A}_i\}_{i\in\{0,1\}}$ and $\boldsymbol{s}$ are chosen uniformly at random, and $\{\mathbf{A}_i\}_{i\in\{0,1\}}$ can be publicly known.

Parameters. Our PRF works for a wide range of parameters. For ease of our security proof, we use a concrete parameter setting following Table 1: Let κ be the security parameter, we set $n = 50\kappa, k = \kappa, p = \kappa, q = \kappa^6$.

Theorem 3.6. *Let κ be security parameter, n, k, p, q be parameters setting above, and χ be a β-bounded distribution over $\mathbb{Z}_q$ for $\beta = \sqrt{\kappa}$. Assume $\mathsf{LWE}_{\ell,2n,q,\chi}$ is (t, ε)-hard where $\ell = \kappa$. Then the family $\mathcal{F}$ constructed above is a $(t^{'}, Q, \varepsilon^{'})$-$\mathsf{PRF}$, where $t^{'} = t - \mathsf{poly}(\kappa), \varepsilon^{'} \leq 48kn\varepsilon + \frac{1}{2^\kappa}$ for sufficient large κ and any $Q = \mathsf{poly}(\kappa)$.*

Proof Sketch. As discussed in the introduction, the proof follows the steps $\mathsf{LWE} \xrightarrow{(i)} n\text{-}\mathsf{LWE} \xrightarrow{(ii)} Q\text{-}\mathsf{LWR}' \xrightarrow{(iii)} \mathsf{PRF}$. Step (i) follows from a standard hybrid argument; Step (ii) follows from Corollary 3.4 in Sect. 3.1; Step (iii) is very similar to the classic proof Q-$\mathsf{PRG} \to \mathsf{PRF}$ (see [12,31,37]). For completeness, we present the formal arguments in full version of this paper.

We can further improve the result by applying the domain extension techniques by [26], resulting in the Corollary as follows:

Corollary 3.7. *Let κ be security parameter, $n = 50\kappa, p = \kappa, q = \kappa^6, k = \kappa, \ell = \kappa, \beta = \sqrt{\kappa}$ as our setting of parameters. We have the following:*

- *Assume n-$\mathsf{LWE}_{\ell,2n,q,\chi}$ is (t,ε)-hard where χ is a β-bounded distribution over $\mathbb{Z}_q$. Then there exists a $(t^{'},Q,\varepsilon^{'})$-$\mathsf{PRF}$, where $t^{'}=t-\mathsf{poly}(\kappa),\varepsilon^{'}\leq\omega(\log\kappa)\varepsilon+2^{-\Omega(\kappa)}$ for sufficient large κ and for any $Q=\mathsf{poly}(\kappa)$.*
- *Assume $\mathsf{LWE}_{\ell,2n,q,\chi}$ is (t,ε)-hard where χ is a β-bounded distribution over $\mathbb{Z}_q$. Then there exists a $(t^{'},Q,\varepsilon^{'})$-$\mathsf{PRF}$, where $t^{'}=t-\mathsf{poly}(\kappa),\varepsilon^{'}\leq 48\kappa\omega(\log\kappa)\varepsilon$ $+2^{-\Omega(\kappa)}$ for sufficient large κ and any $Q=\mathsf{poly}(\kappa)$.*

4 New Framework of Lattice-Based IBE with Tight Security Under poly Modulus

In this section, we propose a novel framework that integrates key homomorphic evaluation on the public matrices, homomorphic evaluation on leveled HE ciphertexts, bootstrapping, and our almost tight PRF in Sect. 3.2. By applying this technique, we construct an almost tight adaptively secure IBE from LWE with a polynomial modulus. Our technique can also apply to the lattice based signature scheme resulting an almost tight security under poly modulus. Due to the space, we put the construction in full version of this paper. We present our IBE construction in Sect. 4.1, and then show the tight security in Sect. 4.2, finally instantiate all the building blocks in Sect. 4.3.

4.1 IBE Construction

- $\mathsf{Setup}(1^\kappa)$ The setup algorithm takes as input a security parameter κ, It does the following:
  1. Sample a random matrix $\mathbf{A}\in\mathbb{Z}_q^{n\times m}$ along with a trapdoor basis $\mathbf{T_A}\in\mathbb{Z}^{m\times m}$ of lattice $\Lambda_q^{\perp}(\mathbf{A})$ by running TrapGen.
  2. Select random matrices $\mathbf{A}_0,\mathbf{A}_1\in\mathbb{Z}_q^{n\times m}$. Run HE.KeyGen algorithm of a HE scheme $\boxed{(\mathsf{ek},\mathsf{evk},\mathsf{dk})\leftarrow\mathsf{HE.KeyGen}}$. Set the random "PRF key" elements as $\boxed{\{\boldsymbol{d}_i\}_{i\in[k_1]}}$ where $\boxed{\boldsymbol{d}_i\xleftarrow{\$}\mathsf{HE.Enc}(\mathsf{ek},0)}$ and set "bootstrapping key" element as $\boxed{\mathsf{evk}}$. Select random "PRF input" elements
  $$\boxed{\boldsymbol{c}_0\xleftarrow{\$}\mathsf{HE.Enc}(\mathsf{ek},0),\boldsymbol{c}_1\xleftarrow{\$}\mathsf{HE.Enc}(\mathsf{ek},1)}$$
  uniformly at random. Select random matrices $\{\mathbf{D}_i\}_{i\in[k_2]}\in\mathbb{Z}_q^{n\times m}$. Express the decryption algorithm HE.Dec as a NAND Boolean circuit $\boxed{C_{\mathsf{Dec}}}$.
  3. Select a random vector $\boldsymbol{u}\xleftarrow{\$}\mathbb{Z}_q^n$.
  4. Select a secure pseudorandom function $\mathsf{PRF}:\{0,1\}^{k_1}\times\{0,1\}^{\ell}\rightarrow\{0,1\}$, express it as a NAND Boolean circuit C_{PRF} with depth $d=d(\kappa)$, and select a PRF key $K=s_1s_2...s_{k_1}\xleftarrow{\$}\{0,1\}^{k_1}$.
  5. Set $\mathsf{msk}=(\mathbf{T_A},K)$, and output
  $$\boxed{\mathsf{mpk}=(\mathbf{A},\{\mathbf{A}_0,\mathbf{A}_1\},\{\boldsymbol{d}_i\}_{i\in[k_1]},\{\mathbf{D}_i\}_{i\in[k_2]},\{\boldsymbol{c}_0,\boldsymbol{c}_1\},\mathsf{evk},\boldsymbol{u},\mathsf{PRF},C_{\mathsf{PRF}}).}$$

- $\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$ The key generation algorithm take $\mathsf{mpk}, \mathsf{msk}$ and an identity $\mathsf{id} = x_1 x_2 ... x_\ell \in \{0,1\}^\ell$ as input, and does the following:
  1. Compute $b = \mathsf{PRF}(K, id)$.
  2. Compute $\boxed{\mathsf{ct_{id}} = \mathsf{HE.Eval}(\mathsf{evk}, C_{\mathsf{PRF}}, (\{\boldsymbol{d}_i\}_{i\in[k_1]}, \{\boldsymbol{c}_{x_i}\}_{i\in[\ell]}))}$.
  3. Compute $\boxed{\mathbf{A}_{C_{\mathsf{PRF}},\mathsf{id}} = \mathsf{Eval}^{\mathsf{Pub}}(C_{\mathsf{Dec}}, (\{\mathbf{D}_i\}_{i\in[k_2]}, \{(\mathsf{ct_{id}})_i \mathbf{G}\}_{i\in[k_3]}))}$, where $\boxed{(\mathsf{ct_{id}})_i}$ is the i-bit of $\mathsf{ct_{id}}$.
  4. Set $\mathbf{F}_{\mathsf{id},1-b} = [\mathbf{A} | \mathbf{A}_{1-b} - \mathbf{A}_{C_{\mathsf{PRF}},\mathsf{id}}] \in \mathbb{Z}_q^{n\times 2m}$.
  5. Run $\mathsf{SampleLeft}$ to sample $\boldsymbol{d}_{\mathsf{id}}$ from the discrete Gaussian distribution $D_{\Lambda_q^{\boldsymbol{u}}(\mathbf{F}_{\mathsf{id},1-b}),s}$, then $\mathbf{F}_{\mathsf{id},1-b}\boldsymbol{d}_{\mathsf{id}} = \boldsymbol{u} (\mathrm{mod}\ q)$. Output $\mathsf{sk_{id}} = (b, \boldsymbol{d}_{\mathsf{id}})$.

- $\mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, \mu)$ To encrypt a message $\mu \in \{0,1\}$ with respect to an identity $\mathsf{id} = x_1 x_2 ... x_\ell \in \{0,1\}^\ell$:
  1. Compute $\boxed{\mathsf{ct_{id}} = \mathsf{HE.Eval}(\mathsf{evk}, C_{\mathsf{PRF}}, (\{\boldsymbol{d}_i\}_{i\in[k_1]}, \{\boldsymbol{c}_{x_i}\}_{i\in[\ell]}))}$.
  2. Compute $\boxed{\mathbf{A}_{C_{\mathsf{PRF}},\mathsf{id}} = \mathsf{Eval}^{\mathsf{Pub}}(C_{\mathsf{Dec}}, (\{\mathbf{D}_i\}_{i\in[k_2]}, \{(\mathsf{ct_{id}})_i \mathbf{G}\}))}$.
  3. Set $\mathbf{F}_{\mathsf{id},b} = [\mathbf{A} | \mathbf{A}_b - \mathbf{A}_{C_{\mathsf{PRF}},\mathsf{id}}] \in \mathbb{Z}_q^{n\times 2m}$ for $b = 0, 1$.
  4. Select two random vectors $\boldsymbol{s}_0, \boldsymbol{s}_1 \xleftarrow{\$} \mathbb{Z}_q^n$.
  5. Select two noise scalars $v_{0,0}, v_{1,0} \leftarrow D_{\mathbb{Z},\sigma_{\mathsf{LWE}}}$ and two noise vectors $\boldsymbol{v}_{0,1}, \boldsymbol{v}_{1,1} \leftarrow D_{\mathbb{Z}^{2m},\sigma}$, where σ is a gaussian parameter lager than σ_{LWE}.
  6. Compute the ciphertext $\mathsf{ct_{id}} = (c_{0,0}, \boldsymbol{c}_{0,1}, c_{1,0}, \boldsymbol{c}_{1,1})$ as:

$$\begin{cases} c_{0,0} = (\boldsymbol{s}_0^t \boldsymbol{u} + v_{0,0} + \mu \lfloor q/2 \rfloor) \bmod q \\ \quad\quad \boldsymbol{c}_{0,1}^t = (\boldsymbol{s}_0^t \mathbf{F}_{\mathsf{id},0} + \boldsymbol{v}_{0,1}^t) \bmod q \end{cases}$$

$$\begin{cases} c_{1,0} = (\boldsymbol{s}_1^t \boldsymbol{u} + v_{1,0} + \mu \lfloor q/2 \rfloor) \bmod q \\ \quad\quad \boldsymbol{c}_{1,1}^t = (\boldsymbol{s}_1^t \mathbf{F}_{\mathsf{id},1} + \boldsymbol{v}_{1,1}^t) \bmod q \end{cases}$$

- $\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk_{id}}, \mathsf{ct_{id}})$ The decryption algorithm uses the key $(b, \boldsymbol{d}_{\mathsf{id}})$ to decrypt $(c_{b,0}, \boldsymbol{c}_{b,1})$. The decryption algorithm computes $\eta = (c_{b,0} - \boldsymbol{c}_{b,1}^t \boldsymbol{d}_{\mathsf{id}}) \bmod q$. If η is closer to 0 that $\pm q/2$, then decryption algorithm outputs $\mu = 0$, otherwise, outputs $\mu = 1$.

Correctness analysis can be verified in the same way as [16]. We omit it here due to the space limit.

Parameter Setting. We now provide an instantiation that achieves both correctness a and security (Table 2).

- To ensure the condition of $\mathsf{TrapGen}$ in Theorem 2.8 and achieve the statistical distance in Lemma 4.2, we set $m = O(n \log q)$, $n \geq \kappa + \log k_2 + 5$;
- According to [3,18,21,30], there exists an HE scheme such that the decryption circuit is in NC_1, so we set $L = O(\log n)$;

Table 2. Parameter setting of IBE scheme

Parameters	Description	Setting
κ	Security parameter	
k_1	Secret key length of PRF	κ
k_2	The length of decryption key of HE	κ
k_3	Output length of HE.Eval	κ
n	Row dimension of public matrix	$\geq 2\kappa + 5$
m	Column dimension of public matrix	$O(n \log q)$
ℓ	Length of id	κ
L	Depth of HE decryption circuit	$O(\log n)$
s	Gaussian parameter of secret key	$O(n^{3+\epsilon})$
σ_{LWE}	Gaussian parameter of LWE error	$O(\sqrt{\kappa + \log \kappa})$
σ	Gaussian parameter of noise vectors in $\boldsymbol{c}_{b,1}$	$2\sigma^* \cdot \sigma_{\mathsf{LWE}}$
σ^*	Parameter of ReRand algorithm	$O(n^{2+\epsilon})$
q	Modulus of LWE	$O(n^{8+\epsilon})$

– To ensure that SampleLeft in the real scheme and SampleRight in the simulation game have the statistical distance within $2^{-(\kappa+2)}/3Q_{\mathsf{id}}$ per Theorem 2.8 and Theorem 2.9, we need

$$s > \|\tilde{\mathbf{T}}_{\mathbf{A}}\| \cdot \omega(\sqrt{\log 2m}) \textit{ and } s > \|\tilde{\mathbf{T}}_{\mathbf{G}}\| \cdot \|\mathbf{R}\| \cdot \omega(\sqrt{\log m}),$$

where $\mathbf{R} = \mathbf{R}_{\mathbf{A}_{1-b}} - \mathbf{R}_{C_{\mathsf{PRF}},\mathsf{id}}$, and $n \geq \kappa + 5 + \log Q_{\mathsf{id}}$ (Q_{id} is number of key queries). According to Theorem 2.14 and the bootstrapping computation [3], the key-homomorphic evaluation algorithm of HE decryption circuit is $O(n^{2+\epsilon})$-compatible for any $\epsilon \in (0,1)$, which means that $\|\mathbf{R}_{C_{\mathsf{PRF}},\mathsf{id}}\| \leq O(n^{2+\epsilon})$. To satisfy these conditions, we set $s = O(n^{3+\epsilon})$ and $n \geq 2\kappa + 5$ (without loss of generality, we assume $Q_{\mathsf{id}} < 2^{\kappa}$);

– To ensure Regev's quantum reduction to LWE [49], we need $\sigma_{\mathsf{LWE}} > 2\sqrt{\kappa}$;

– For ReRand algorithm to work with the statistical distance in Lemma 4.3, we need $\sigma^* > \mathsf{s}_1([\mathbf{I}|\mathbf{R}])$, $\sigma_{\mathsf{LWE}} > max\{\eta_\epsilon(\mathbb{Z}^m), \eta_\epsilon(\mathbb{Z}^\ell)\}$ and $\sigma = 2\sigma^* \cdot \sigma_{\mathsf{LWE}}$. According to the property of smoothing parameters (which can be found in full version of this paper) and Theorem 2.14, we set $\sigma_{\mathsf{LWE}} = O(\sqrt{\kappa + \log \kappa}), \sigma^* = O(n^{2+\epsilon})$;

– To ensure the correctness of decryption, we need $|c_{b,0} - \boldsymbol{c}_{b,1}^t \boldsymbol{d}_{\mathsf{id}}| < q/4$, as a result $O(s \cdot m \cdot \sigma) < q/4$. We set $q = O(n^{8+\epsilon})$ (q is not necessarily a prime).

4.2 Security

The security of the IBE scheme above can be stated by the following theorem.

Theorem 4.1. *Let the parameters be chosen as above, and χ be the distribution $\mathcal{D}_{\mathbb{Z}^m,\sigma_{\mathsf{LWE}}}$. If the $\mathsf{LWE}_{n,m,q,\chi}$ problem is $(t_{\mathsf{LWE}}, \varepsilon_{\mathsf{LWE}})$-hard, HE scheme is $(t_{\mathsf{HE}}, k_1, \varepsilon_{\mathsf{HE}})$-IND secure with decryption circuit in $\mathbf{NC}_1$ (e.g., $O(n^{2+\epsilon})$-compatible), and the PRF used in the IBE is a $(t_{\mathsf{PRF}}, Q_{\mathsf{id}}, \varepsilon_{\mathsf{PRF}})$-PRF, then the IBE scheme constructed above is $(t^*, Q_{\mathsf{id}}, \varepsilon^*)$-adaptively secure such that $\varepsilon^* \leq 2(\varepsilon_{\mathsf{LWE}} + \varepsilon_{\mathsf{PRF}}) + 3\varepsilon_{\mathsf{HE}} + 2^{-\kappa}$, and $t^* = \min\{T_{\mathsf{LWE}}, T_{\mathsf{PRF}}, T_{\mathsf{HE}}\} - \mathsf{poly}(n, m, k, Q_{\mathsf{id}}, \log q)$.*

Proof. We prove the theorem by a sequence of hybrid games. Given a PPT adversary $\mathcal{A}$, the first game is defined as the real adaptive security game. Then we will show that all the neighboring games are computationally/statistically indistinguishable. Finally we show that $\mathcal{A}$ has no advantage in the last game to complete the proof.

Before we present the hybrids, we first define the following simulation algorithms Sim.Setup, Sim.KeyGen and Sim.Enc, making essential modifications of those in the work Boyen and Li [16]. We highlight the differences in boxes.

- $\mathsf{Sim.Setup}(1^\kappa)$ The algorithm does the following:
  1. Select a matrix $\mathbf{A} \overset{\$}{\leftarrow} \mathbb{Z}_q^{n\times m}$. Run HE.KeyGen algorithm of a HE scheme $\boxed{(\mathsf{ek}, \mathsf{evk}, \mathsf{dk}) \leftarrow \mathsf{HE.KeyGen}}$. Set "bootstrapping key" element as $\boxed{\mathsf{evk}}$. Select random "PRF input" elements
  $$\boxed{\boldsymbol{c}_0 \overset{\$}{\leftarrow} \mathsf{HE.Enc}(\mathsf{ek}, 0)}, \boxed{\boldsymbol{c}_1 \overset{\$}{\leftarrow} \mathsf{HE.Enc}(\mathsf{ek}, 1)}$$
  uniformly at random. Express the decryption circuit HE.Dec as a NAND Boolean circuit $\boxed{C_{\mathsf{Dec}}}$ and express dk as $\boxed{\mathsf{dk} = (dk_1, ..., dk_{k_2})}$.
  2. Select $k_2 + 2$ low-norm matrices $\boxed{\{\mathbf{R}_{\mathbf{A}_b}\}_{b\in\{0,1\}}, \{\mathbf{R}_{\mathbf{D}_i}\}_{i\in[k_2]} \overset{\$}{\leftarrow} \{0,1\}^{m\times m}}$.
  3. Select a secure $\mathsf{PRF} : \{0,1\}^{k_1} \times \{0,1\}^{\ell} \to \{0,1\}$ and express it as a NAND Boolean circuit C_{PRF} with depth $d = d(\kappa)$.
  4. Select a uniformly random string $K = s_1 s_2 ... s_{k_1} \overset{\$}{\leftarrow} \{0,1\}^{k_1}$.
  5. Set $\mathbf{A}_b = \mathbf{A}\mathbf{R}_{\mathbf{A}_b} + b\mathbf{G}$ for $b = 0, 1$ and $\boxed{\mathbf{D}_i = \mathbf{A}\mathbf{R}_{\mathbf{D}_i} + dk_i\mathbf{G}}$ for $i \in [k_2]$.
  6. Set the random "PRF key" elements as $\boxed{\{\boldsymbol{d}_i\}_{i\in[k_1]}}$ where
  $\boxed{\boldsymbol{d}_i \overset{\$}{\leftarrow} \mathsf{HE.Enc}(\mathsf{ek}, s_i)}$.
  7. Set vector $\boldsymbol{u} \overset{\$}{\leftarrow} \mathbb{Z}_q^n$, and publish
  $$\boxed{\mathsf{mpk} = (\mathbf{A}, \{\mathbf{A}_0, \mathbf{A}_1\}, \{\boldsymbol{d}_i\}_{i\in[k_1]}, \{\boldsymbol{c}_0, \boldsymbol{c}_1\}, \{\mathbf{D}_i\}_{i\in[k_2]}, \mathsf{evk}, \boldsymbol{u}, \mathsf{PRF}, C_{\mathsf{PRF}})}.$$
- $\mathsf{Sim.KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$ Upon an input identity $\mathsf{id} = x_1 x_2 ... x_\ell \in \{0,1\}^\ell$, the algorithm uses mpk, msk to do the following:
  1. Compute $\boxed{\mathsf{ct}_{\mathsf{id}} = \mathsf{HE.Eval}(\mathsf{evk}, C_{\mathsf{PRF}}, (\{\boldsymbol{d}_i\}_{i\in[k_1]}, \{\boldsymbol{c}_{x_i}\}_{i\in[\ell]}))}$ and
  $$\boxed{\mathbf{R}_{C_{\mathsf{PRF}},\mathsf{id}} = \mathsf{Eval}^{\mathsf{Trap}}(C_{\mathsf{Dec}}, \mathbf{A}, (\{dk_i\}_{i\in[k_2]}, \{(\mathsf{ct}_{\mathsf{id}})_i\}_{i\in[k_3]}), (\{\mathbf{R}_{\mathbf{D}_i}\}_{i\in[k_2]}, \{[0]_i\}_{i\in[k_3]}))},$$
  where for each $i \in [k_3]$, $[0]_i$ denotes 0 matrix with dimension $m \times m$.
  2. Let $\mathsf{PRF}(K, \mathsf{id}) = b \in \{0,1\}$. Set
  $$\mathbf{F}_{\mathsf{id},1-b} = [\mathbf{A}|\mathbf{A}_{1-b} - \mathbf{A}_{C_{\mathsf{PRF}},\mathsf{id}}] = [\mathbf{A}|\mathbf{A}(\mathbf{R}_{\mathbf{A}_{1-b}} - \mathbf{R}_{C_{\mathsf{PRF}},\mathsf{id}}) + (1-2b)\mathbf{G}].$$
  3. Run SampleRight to sample $\boldsymbol{d}_{\mathsf{id}} \in D_{\Lambda_q^{\boldsymbol{u}}(\mathbf{F}_{\mathsf{id},1-b}),s}$, and output $\boxed{\mathsf{sk}_{\mathsf{id}} = (b, \boldsymbol{d}_{\mathsf{id}})}$.

- $\mathsf{Sim.Enc}(\mathsf{mpk}, \mathsf{id}^*, \mu)$ The algorithm takes a message μ, mpk and a challenge identity id^* as input, does the following:
  1. Compute $b = \mathsf{PRF}(K, \mathsf{id}^*)$.
  2. Set $\mathbf{F}_{\mathsf{id}^*,b} = [\mathbf{A}|\mathbf{A}_b - \mathbf{A}_{C_{\mathsf{PRF}},\mathsf{id}^*}] = [\mathbf{A}|\mathbf{A}(\mathbf{R}_{\mathbf{A}_b} - \mathbf{R}_{C_{\mathsf{PRF}},\mathsf{id}^*})]$. and
  $$\mathbf{F}_{\mathsf{id}^*,1-b} = [\mathbf{A}|\mathbf{A}_{1-b} - \mathbf{A}_{C_{\mathsf{PRF}},\mathsf{id}^*}] = [\mathbf{A}|\mathbf{A}(\mathbf{R}_{\mathbf{A}_{1-b}} - \mathbf{R}_{C_{\mathsf{PRF}},\mathsf{id}^*}) + (1-2b)\mathbf{G}].$$
  3. Select random vectors $\boldsymbol{s}_b, \boldsymbol{s}_{1-b} \xleftarrow{\$} \mathbb{Z}_q^n$.
  4. Select noise scalars $v_{b,0}, v_{1-b,0} \leftarrow D_{\mathbb{Z},\sigma_{\mathsf{LWE}}}$, and noise vectors $\boxed{\boldsymbol{v}'_{b,1} \leftarrow D_{\mathbb{Z}^m,\sigma_{\mathsf{LWE}}}}$.
  5. Let $\mathbf{R} = \mathbf{R}_{\mathbf{A}_b} - \mathbf{R}_{C_{\mathsf{PRF}},\mathsf{id}^*}$, and set $\boxed{\sigma^* = \sigma/2\sigma_{\mathsf{LWE}}}$. Then invoke the ReRand algorithm to compute
  $$\boxed{\boldsymbol{v}_{b,1} = \mathsf{ReRand}([\mathbf{I}|\mathbf{R}], \boldsymbol{s}_b^t\mathbf{A} + \boldsymbol{v}'_{b,1}, \sigma_{\mathsf{LWE}}, \sigma^*) - \mathbf{F}_{\mathsf{id}^*,b}^t\boldsymbol{s}_b}.$$
  6. Select noise vectors $\boldsymbol{v}_{1-b,1} \leftarrow D_{\mathbb{Z}^{2m},\sigma}$.
  7. Set the challenge ciphertext $\mathsf{ct}_{\mathsf{id}^*} = (c_{b,0}, \boldsymbol{c}_{b,1}, c_{1-b,0}, \boldsymbol{c}_{1-b,1})$ as:
  $$\begin{cases} c_{b,0} = \left(\boldsymbol{s}_b^t\boldsymbol{u} + v_{b,0} + \mu\lfloor q/2\rfloor\right) \bmod q \\ \boldsymbol{c}_{b,1}^t = \left(\boldsymbol{s}_b^t\mathbf{F}_{\mathsf{id}^*,b} + \boldsymbol{v}_{b,1}^t\right) \bmod q \end{cases}$$
  $$\begin{cases} c_{1-b,0} = \left(\boldsymbol{s}_{1-b}^t\boldsymbol{u} + v_{1-b,0} + \mu\lfloor q/2\rfloor\right) \bmod q \\ \boldsymbol{c}_{1-b,1}^t = \left(\boldsymbol{s}_{1-b}^t\mathbf{F}_{\mathsf{id}^*,1-b} + \boldsymbol{v}_{1-b,1}^t\right) \bmod q \end{cases}$$

Now we present a sequence of games and prove that the neighboring games are indistinguishable. We follow the structure of the sequence from Boyen and Li [16], and add an additional step to incorporate the homomorphic encryption.

Game 0: This is the real adaptive security game, and all the algorithms are the same as the real game.

Game 1: This game is the same as **Game 0** except it runs $\mathsf{Sim.Setup}$ and $\mathsf{Sim.KeyGen}$ instead of Setup and KeyGen.

Game 2: This game is the same as **Game 1** except that the challenge ciphertext is generated by $\mathsf{Sim.Enc}$ rather than Enc.

Game 3: This game is the same as **Game 2** except that during the generation of challenge ciphertext, it samples $(c_{b,0}, \boldsymbol{c}_{b,1})$ uniformly random from $\mathbb{Z}_q \times \mathbb{Z}_q^{2m}$ for $b = \mathsf{PRF}(K, \mathsf{id}^*)$, and $(c_{1-b,0}, \boldsymbol{c}_{1-b,1})$ is computed by $\mathsf{Sim.Enc}$ as in **Game 2**.

Game 4: This game is the same as **Game 3** except for $b = \mathsf{PRF}(K, \mathsf{id}^*)$ it runs Enc to generate $(c_{1-b,0}, \boldsymbol{c}_{1-b,1})$ instead of using $\mathsf{Sim.Enc}$.

Game 5: This game is the same as **Game 4** except it runs Setup and KeyGen to generate mpk and $\mathsf{sk}_{\mathsf{id}^*}$.

Game 6: This game is the same as **Game 5** except that for $b = \mathsf{PRF}(K, \mathsf{id}^*)$, the challenge ciphertext part $(c_{b,0}, \boldsymbol{c}_{b,1})$ is generated by Enc rather than choosing it randomly, and $(c_{1-b,0}, \boldsymbol{c}_{1-b,1})$ is chosen randomly.

Game 7: This game is the same as **Game 6** except that it runs $\mathsf{Sim.Setup}$ and $\mathsf{Sim.KeyGen}$ to generate mpk and $\mathsf{sk}_{\mathsf{id}^*}$.

Game 8: This game is the same as **Game 7** except that for $b = \mathsf{PRF}(K, \mathsf{id}^*)$, it computes the challenge ciphertext $(c_{b,0}, \boldsymbol{c}_{b,1})$ by $\mathsf{Sim.Enc}$.

Game 9: This game is the same as **Game 8** except that the whole challenge ciphertext is sampled uniformly at random. As the challenge ciphertext is independent of the adversary $\mathcal{A}$, clearly in **Game 9** the adversary has no advantage.

We let W_i be the event that $\gamma^{'} = \gamma$ at the end of the **Game** i, and set the advantage's advantage in **Game** i as $|\Pr[W_i] - 1/2|$. We prove the following lemmas, which together imply Theorem 4.1.

Lemma 4.2. ***Game 0*** *and* ***Game 1*** *are* $(T_1, \varepsilon_{\mathsf{HE}} + 2^{-(\kappa+2)})$*-indistinguishable, assuming the* HE *scheme is* $(T_{\mathsf{HE}}, \varepsilon_{\mathsf{HE}})$*-CPA secure, where* $T_1 = T_{\mathsf{HE}} - \mathsf{poly}(n, k, m, Q_{\mathsf{id}}, \log q)$.

Proof. We analyze the only four differences between **Game 0** and **Game 1:**

1. In **Game 0**, the matrix $\mathbf{A}$ is generated by $\mathsf{TrapGen}$, and the matrix $\mathbf{A}$ is chosen uniformly at random in **Game 0**. By Theorem 2.8, these two distributions of constructing matrix $\mathbf{A}$ are statistically close. More precisely, the statistical distance is within $2^{-(\kappa+2)}/3$ by our parameter setting.
2. In **Game 0**, the matrices $\{\mathbf{A}_0, \mathbf{A}_1\}$ are chosen uniformly at random from $\mathbb{Z}_q^{n\times m}$. While in **Game 1**, these matrices are computed as $\mathbf{A}_b = \mathbf{A}\mathbf{R}_{\mathbf{A}_b} + b\mathbf{G}$, for $b \in \{0, 1\}$ for random low-norm matrices $\{\mathbf{R}_{\mathbf{A}_b}\}_{b\in\{0,1\}}$ from $\{0, 1\}^{m\times m}$. By Theorem 3.1, the distributions of these matrices in the two games are statistically close. More precisely, the statistical distance is within $2^{-(\kappa+1)}/(3k_2+6)$ by our parameter setting.
3. In **Game 0**, the elements $\{\boldsymbol{d}_i\}_{i\in[k_1]}$ are k_1 ciphertexts $\mathsf{HE.Enc}(\mathsf{pk}, 0)$ and $\{\mathbf{D}_i\}_{i\in[k_2]}$ are chosen uniformly at random from $\mathbb{Z}_q^{n\times m}$. In **Game 1**, these elements are the ciphertexts $\mathsf{HE.Enc}(\mathsf{pk}, s_i)$ and $\{\mathbf{D}_i\}_{i\in[k_2]}$ are the matrices $\mathbf{D}_i = \mathbf{A}\mathbf{R}_{\mathbf{D}_i} + t_i\mathbf{G}$. We show the indistinguishability of the two cases by bybrid argument, we define a sequence of sub-hybirds:
   - H_0: Sample $\{\mathbf{d}_i\}_{i\in[k_1]}$ and $\{\mathbf{D}_i\}_{i\in[k_2]}$ as in **Game 0**.
   - H_1: Generate $\{\mathbf{d}_i\}_{i\in[k_1]}$ as in **Game 1**. Set $\{\mathbf{D}_i\}_{i\in[k_2]}$ as in **Game 0**.
   - H_2: Set $\{\mathbf{d}_i\}_{i\in[k_1]}$ and $\{\mathbf{D}_i\}_{i\in[k_2]}$ as in **Game 1**.

   We first show that the neighboring games H_0 and H_1 are $(T^{'}, \varepsilon_{\mathsf{HE}})$-indistinguishable by assuming that HE scheme is $(T_{\mathsf{HE}}, \varepsilon_{\mathsf{HE}})$-secure, where $T^{'} = T_{\mathsf{HE}} - \mathsf{poly}(n, m, k, \log q)$. Then, we show that H_1 and H_2 are statistically close by Theorem 3.1.

   Without loss of generality, if there exists a distinguisher $\mathcal{D}$ can distinguish H_0 from H_1 within running time $T_{\mathcal{D}} \leq T'$ and with advantage $\varepsilon_{\mathcal{D}} \geq \varepsilon_{\mathsf{HE}}$, then we construct a reduction $\mathcal{B}$ that breaks HE as follows:
   - $\mathcal{B}$ chooses $\{\mathbf{D}_i\}_{i\in[k_2]}$ uniformly at random from $\mathbb{Z}_q^{n\times m}$.

- $\mathcal{B}$ sets $\boldsymbol{m}_0 = (s_1, ..., s_{k_1}), \boldsymbol{m}_1 = (0, ...0)$ as its challenge messages, and forwards $\boldsymbol{m}_0, \boldsymbol{m}_1$ to the challenger. $\mathcal{B}$ gets the challenge ciphertext $\mathsf{ct}^* = \{\mathsf{ct}_i\}_{i\in[k_1]}$ from the challenger, and sets $\mathsf{ct}^* = \{\boldsymbol{d}_i\}_{i\in[k_1]}$.
- $\mathcal{B}$ simulates the hybrid game (either H_0 or H_1) with $\{\boldsymbol{d}_i\}_{i\in[k_1]}, \{\mathbf{D}_i\}_{i\in[k_2]}$ and then outputs the outcome of $\mathcal{D}$.

Clearly, if the challenger encrypts $\boldsymbol{m}_0$, then $\mathcal{B}$ simulates the hybrid H_0, and otherwise, the hybrid H_1. Therefore, $\mathcal{B}$ has the same advantage as $\mathcal{D}$, i.e., $\varepsilon_{\mathcal{D}} \geq \varepsilon_{\mathsf{HE}}$, in breaking HE, and the running time of $\mathcal{B}$ is within $T_{\mathcal{D}} + \mathsf{poly}(n, m, k, \log q) \leq T_{\mathsf{HE}}$. This is a contradiction to the security of HE.

The difference between H_1 and H_2 is the generation of the matrices $\{\mathbf{D}_i\}_{i\in[k_2]}$. By Theorem 3.1, $\{\mathbf{D}_i\}_{i\in[k_2]}$ in the two cases are statistically close, and more precisely, the statistical distance of H_1 and H_2 is within $k_2 \times 2^{-(\kappa+2)}/(3k_2+6)$ by our setting of parameters.

4. In both **Game 0** and **Game 1**, the use of $\mathbf{A}_0$ or $\mathbf{A}_1$ in the key generation algorithms is decided by $b = \mathsf{PRF}(K, \mathsf{id})$. For a private key query on id in **Game 1**, let

$$\mathbf{F}_{\mathsf{id},1-b} = [\mathbf{A}|\mathbf{A}_{1-b} - \mathbf{A}_{C_{\mathsf{PRF}},\mathsf{id}}] = [\mathbf{A}|\mathbf{A}\cdot(\mathbf{R}_{\mathbf{A}_{1-b}} - \mathbf{R}_{C_{\mathsf{PRF}},\mathsf{id}}) + (1-2b)\mathbf{G}].$$

Note that the trapdoor of $\Lambda_q^{\perp}(\mathbf{G})$ is also a trapdoor of $\Lambda_q^{\perp}((1-2b)\mathbf{G})$. In **Game 0**, $\boldsymbol{d}_{\mathsf{id}}$ is generated by $\mathsf{SampleLeft}$ with the trapdoor $\mathbf{T}_{\mathbf{A}}$. In **Game 1**, $\boldsymbol{d}_{\mathsf{id}}$ is generated by $\mathsf{SampleRight}$ with the trapdoor of $\Lambda_q^{\perp}((1-2b)\mathbf{G})$. By Theorem 2.9 and our setting of parameters, the statistical distance between the distributions of a single key $\boldsymbol{d}_{\mathsf{id}}$ in the two cases is bounded by $2^{-(\kappa+2)}/3Q_{\mathsf{id}}$. Therefore, from a simple union bound over Q_{id} keys, we conclude that the secret key distributions generated in these two ways are within a statistical distance up to $2^{-(\kappa+2)}/3$.

By combining the arguments above, we conclude that **Game 0** and **Game 1** are $(T_1, \varepsilon_{\mathsf{HE}} + 2^{-(\kappa+2)})$-indistinguishable, where $T_1 = T_{\mathsf{HE}} - \mathsf{poly}(n, m, k, \log q)$. □

Lemma 4.3. *Game 1 and Game 2 are* $(\infty, 2^{-(\kappa+2)}/2)$*-indistinguishable.*

Proof. The only difference between **Game 1** and **Game 2** is the way how the challenge ciphertext is generated. Particularly, in **Game 1**, the challenge ciphertext is generated by Enc, and the noise vectors are sampled from some discrete Gaussian distributions that are independent of mpk. In **Game 2** the challenge ciphertext is generated by $\mathsf{Sim.Enc}$.

By construction, Enc and $\mathsf{Sim.Enc}$ generate $(c_{b,0}, c_{1-b,0}, \boldsymbol{c}_{1-b,1})$ in the same way, so the distributions of $(c_{b,0}, c_{1-b,0}, \boldsymbol{c}_{1-b,1})$ are identical for the two cases.

By the construction of $\boldsymbol{c}_{b,1}$ in the challenge ciphertext in **Game 2**,

$$\begin{aligned}
\boldsymbol{c}_{b,1}^t &= \left(\boldsymbol{s}_b^t \mathbf{F}_{\mathsf{id}^*,b} + \boldsymbol{v}_{b,1}^t\right) \bmod q \\
&= \left(\boldsymbol{s}_b^t[\mathbf{A}|\mathbf{A}(\mathbf{R}_{\mathbf{A}_b} - \mathbf{R}_{C_{\mathsf{PRF}},\mathsf{id}^*})] + \mathsf{ReRand}([\mathbf{I}|\mathbf{R}], \boldsymbol{s}_b^t\mathbf{A} + \boldsymbol{v}_{b,1}^{'}, \sigma_{\mathsf{LWE}}, \sigma^*)\right) \bmod q \\
&= \left(\boldsymbol{s}_b^t[\mathbf{A}|\mathbf{AR}] + \mathsf{ReRand}([\mathbf{I}|\mathbf{R}], \boldsymbol{s}_b^t\mathbf{A} + \boldsymbol{v}_{b,1}^{'}, \sigma_{\mathsf{LWE}}, \sigma^*)\right) \bmod q.
\end{aligned}$$

It is easy to see that the elements $\boldsymbol{s}_b, \mathbf{A}, \mathbf{R}, \boldsymbol{v}_{b,1}^t$ appearing in the ciphertext of **Game 2** have the same distributions as those in **Game 1**. The only difference is the generation of $\boldsymbol{v}_{b,1}$. In **Game 1**, $\boldsymbol{v}_{b,1}$ is sampled from $D_{\mathbb{Z}^{2m},\sigma}$. In **Game 2**, $\boldsymbol{v}_{b,1}$ is the output of $\mathsf{ReRand}([\mathbf{I}|\mathbf{R}], \boldsymbol{s}_b^t\mathbf{A}+\boldsymbol{v}_{b,1}^{'}, \sigma_{\mathsf{LWE}}, \sigma^*)$, resulting the output gaussian parameter $r = 2\sigma_{\mathsf{LWE}} \cdot \sigma^* = \sigma$. By Lemma 2.13 and our setting of parameters, the statistical distance between the distributions of $\boldsymbol{v}_{b,1}$ in the two cases is bounded by $2^{-(\kappa+2)}/2$. Therefore, the statistical distance between **Game 1** and **Game 2** is bounded by $2^{-(\kappa+2)}/2$. □

Lemma 4.4. ***Game 2** and **Game 3** are $(T_3, \varepsilon_{\mathsf{LWE}})$-indistinguishable, where $T_3 = T_{\mathsf{LWE}} - \mathsf{poly}(n, m, k, Q_{\mathsf{id}}, \log q)$, assuming $\mathsf{LWE}_{n,q,\chi}$ problem is $(T_{\mathsf{LWE}}, \varepsilon_{\mathsf{LWE}})$-hard.*

Proof. We show this by reduction. Assume that there exists a distinguisher $\mathcal{D}$ that distinguishes **Game 2** from **Game 3** within time $T_{\mathcal{D}} \leq T_3$ and with advantage $\varepsilon_{\mathcal{D}} \geq \varepsilon_{\mathsf{LWE}}$, then we construct a $(T_{\mathsf{LWE}}, \varepsilon_{\mathsf{LWE}})$-reduction $\mathcal{B}$ that breaks the LWE assumption. This is a contradiction to the LWE assumption.

The reduction algorithm $\mathcal{B}$ leverages $\mathcal{D}$ to break the the LWE hardness as follows: at the beginning, $\mathcal{B}$ receives the LWE challenge $(\mathbf{A}, \boldsymbol{b}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$ and $(\boldsymbol{a}, b) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, which is either from $\mathcal{O}_\$$ or $\mathcal{O}_{\boldsymbol{s}}$, where $\mathcal{O}_\$$ is the uniformly random distribution over $\mathbb{Z}_q^{n\times(m+1)} \times \mathbb{Z}_q^{m+1}$ and $\mathcal{O}_{\boldsymbol{s}}$ is the distribution of $m+1$ LWE instances with same secret $\boldsymbol{s}$. $\mathcal{B}$ does as follows:

- Setup: Set $\mathbf{A}$ as the public matrix in mpk and $\boldsymbol{a} = \boldsymbol{u}$. Set other public parameters as **Game 2**.
- $\mathsf{Phase\ 1}$: $\mathcal{B}$ answers the secret key queries as **Game 2**.
- $\mathsf{Challenge}$: $\mathcal{B}$ computes the challenge ciphertext of id^* as follows.
  1. Let $b = \mathsf{PRF}(K, \mathsf{id}^*)$. $\mathcal{B}$ sets
  $$\begin{aligned}\mathbf{F}_{\mathsf{id}^*,1-b} &= [\mathbf{A}|\mathbf{A}_{1-b} - \mathbf{A}_{C_{\mathsf{PRF}},\mathsf{id}}] \\ &= [\mathbf{A}|\mathbf{A}(\mathbf{R}_{\mathbf{A}_{1-b}} - \mathbf{R}_{C_{\mathsf{PRF}},\mathsf{id}^*}) + (1-2b)\mathbf{G}].\end{aligned}$$
  2. Let $\mathbf{R} = \mathbf{R}_{\mathbf{A}_b} - \mathbf{R}_{C_{\mathsf{PRF}},\mathsf{id}^*}$. Then constructs $(c_{b,0}, \boldsymbol{c}_{b,1})$ as
  $$\begin{cases} c_{b,0} = (b + \mu\lfloor q/2 \rfloor) \mod q \\ \boldsymbol{c}_{b,1}^t = (\mathsf{ReRand}([\mathbf{I}|\mathbf{R}], \boldsymbol{b}, \sigma_{\mathsf{LWE}}, \sigma^*)) \mod q \end{cases}$$
  3. $\mathcal{B}$ sets $(c_{1-b,0}, \boldsymbol{c}_{1-b,1})$ the same as **Game 2**.
- $\mathsf{Phase\ 2}$: $\mathcal{B}$ replies the secret key queries as in **Game 2**.
- Gauss: If $\mathcal{D}$ outputs "**Game 2**", $\mathcal{B}$ decides that the challenge is from $\mathcal{O}_{\boldsymbol{s}}$. Otherwise, $\mathcal{B}$ decides that the challenge is from $\mathcal{O}_\$$.

If $\mathcal{B}$ gets an LWE instance from the oracle $\mathcal{O}_{\boldsymbol{s}}$, then the distributions of the elements $c_{b,0}, \boldsymbol{c}_{b,1}$ in the challenge ciphertext are the same as in **Game 2**. Therefore, $\mathcal{B}$ simulates **Game 2** for $\mathcal{D}$ in this case. On the other hand, if $\mathcal{B}$ gets an instance from the oracle $\mathcal{O}_\$$, then $c_{b,0}, \boldsymbol{c}_{b,1}$ are uniformly at random, which distribute as the case of **Game 3**. Thus $\mathcal{B}$ simulates **Game 3** in this case. As a result, the advantage of $\mathcal{B}$ is the same as that of $\mathcal{D}$, i.e., $\varepsilon_{\mathcal{D}} \geq \varepsilon_{\mathsf{LWE}}$, and the running time of $\mathcal{B}$ is at most $= T_{\mathcal{D}} + \mathsf{poly}(n, m, k, Q_{\mathsf{id}}, \log q) \leq T_{\mathsf{LWE}}$. This completes the proof. □

Lemma 4.5. ***Game 3*** *and* ***Game 4*** *are identically distributed.*

Proof. It is easy to see that the ways of generating the challenge ciphertext $c_{1-b,0}, \boldsymbol{c}_{1-b,1}$, from Enc and $\mathsf{Sim.Enc}$, are identical. Thus, the advantages of the adversary in **Game 3** and **Game 4** are identical. □

Lemma 4.6. ***Game 4*** *and* ***Game 5*** *are* $(T_5, \varepsilon_{\mathsf{HE}} + 2^{-(\kappa+2)})$*-indistinguishable, assuming* HE *is* $(T_{\mathsf{HE}}, \varepsilon_{\mathsf{HE}})$*-CPA secure, where* $T_5 = T_{\mathsf{HE}} - \mathsf{poly}(n, m, k, Q_{\mathsf{id}}, \log q)$*.*

Proof. The proof is the same as Lemma 4.2. □

Lemma 4.7. ***Game 5*** *and* ***Game 6*** *are* $(T_6, 2\varepsilon_{\mathsf{PRF}})$*-indistinguishable, assuming the* PRF *is* $(T_{\mathsf{PRF}}, \varepsilon_{\mathsf{PRF}})$*-secure, where* $T_6 = T_{\mathsf{PRF}} - \mathsf{poly}(n, m, k, Q_{\mathsf{id}}, \log q)$*.*

Proof. Let $b = \mathsf{PRF}(K, \mathsf{id}^*)$ for the challenge identity id^*. Recall that in **Game 5**, the ciphertext component $(c_{b,0}, \boldsymbol{c}_{b,1})$ is uniformly random and $(c_{1-b,0}, \boldsymbol{c}_{1-b,1})$ is generated by Enc. In **Game 6**, the ciphertext component $(c_{b,0}, \boldsymbol{c}_{b,1})$ is generated by Enc and $(c_{1-b,0}, \boldsymbol{c}_{1-b,1})$ is uniformly random. We prove the indistinguishability between **Game 5** and **Game 6** by three steps.

First we define **Game 5**$^{'}$, which is the same as **Game 5** except that it samples $b \xleftarrow{\$} \{0, 1\}$ to generate the secret keys and challenge ciphertext instead of computing it by PRF. We note that if the same identity is queried multiple times, the same b will be used. Clearly, a distinguisher between **Game 5**$^{'}$ and **Game 5** leads to an attacker for PRF. So **Game 5**$^{'}$ and **Game 5** are $(T_6^{'}, \varepsilon_{\mathsf{PRF}})$-indistinguishable.

Second, we define **Game 5**$^{''}$, which is the same as **Game 5**$^{'}$ except that for randomly sampled b for id^*, it runs Enc to produce $(c_{b,0}, \boldsymbol{c}_{b,1})$ and samples $(c_{1-b,0}, \boldsymbol{c}_{1-b,1})$ uniformly at random. As b is uniformly at random, the advantages of the adversary in **Game 5**$^{''}$ and **Game 5**$^{'}$ are the same.

Finally, because **Game 5**$^{''}$ and **Game 6** are the same except that b is computed via PRF, **Game 5**$^{''}$ and **Game 6** are $(T_6^{'}, \varepsilon_{\mathsf{PRF}})$-indistinguishable.

The lemma follows directly by combining arguments in these three steps. □

Lemma 4.8. ***Game 6*** *and* ***Game 7*** *are* $(T_7, \varepsilon_{\mathsf{HE}} + 2^{-(\kappa+2)})$*-indistinguishable, assuming the* HE *scheme is* $(T_{\mathsf{HE}}, \varepsilon_{\mathsf{HE}})$*-CPA secure, where* $T_7 = T_{\mathsf{HE}} - \mathsf{poly}(n, m, k, Q_{\mathsf{id}}, \log q)$*.*

Proof. The proof is the same as the proof of Lemma 4.2. □

Lemma 4.9. ***Game 7*** *and* ***Game 8*** *are* $(\infty, 2^{-(\kappa+2)}/2)$*-indistinguishable.*

Proof. The proof is the same as the proof for Lemma 4.3. □

Lemma 4.10. ***Game 8*** *and* ***Game 9*** *are* $(T_9, \varepsilon_{\mathsf{LWE}})$*-indistinguishable, assuming* $\mathsf{LWE}_{n,q,\chi}$ *problem is* $(T_{\mathsf{LWE}}, \varepsilon_{\mathsf{LWE}})$*-hard, where* $T_9 = T_{\mathsf{LWE}} - \mathsf{poly}(n, m, k, Q_{\mathsf{id}}, \log q)$*.*

Proof. The proof is the same as the proof for Lemma 4.4. □

By combining all the lemmas above with the composition property of (computational) indistinguishability, we conclude that

$$|\mathsf{Pr}[W_0]-1/2| \leq \sum_{i=0}^{8} |\mathsf{Pr}[W_i]-\mathsf{Pr}[W_{i+1}]| + |\mathsf{Pr}[W_9]-1/2| \leq 2(\varepsilon_{\mathsf{PRF}}+\varepsilon_{\mathsf{LWE}})+3\varepsilon_{\mathsf{HE}}+2^{-\kappa},$$

and

$$\begin{aligned} t^* &= \min\{T_1, T_3, T_5, T_6, T_7, T_9\} - \mathsf{poly}(n, m, k, Q_{\mathsf{id}}, \log q) \\ &= \min\{T_{\mathsf{LWE}}, T_{\mathsf{PRF}}, T_{\mathsf{HE}}\} - \mathsf{poly}(n, m, k, Q_{\mathsf{id}}, \log q). \end{aligned}$$

□

4.3 Instantiations of LWE-based PRF and HE

We point out that all the building blocks can be instantiated under LWE with a polynomial modulus and almost tight analyses. For the PRF, we can use our construction in this work (see Corollary 3.7 in Sect. 3.2). For the homomorphic encryption, we can use the schemes [3,22] (which can be found in full version of this paper). Putting things together, we achieve the following corollary.

Corollary 4.11. *For certain* $n, m, q = \mathsf{poly}(\kappa), \chi$ *such that* $\mathsf{LWE}_{n,m,q,\chi}$ *is* $(t_{\mathsf{LWE}}, \varepsilon_{\mathsf{LWE}})$*-hard, there exists a* $(t^*, Q_{\mathsf{id}}, \varepsilon^*)$*-adaptively secure* IBE*, where* $\varepsilon^* \leq \kappa\omega(\log\kappa)\, \varepsilon_{\mathsf{LWE}} + \mathsf{negl}(\kappa)$ *and* $t^* = t_{\mathsf{LWE}} - \mathsf{poly}(n, m, Q_{\mathsf{id}}, \log q)$*, for any polynomial* Q_{id}*.*

5 ABM-LTF with Tight Security Under poly Modulus

In this section, we present a new construction of almost tight ABM-LTF based on LWE with a polynomial modulus. This improves the work of Libert et al. [41], which requires a super-polynomial modulus. The crux of our improvement relies on our new insight as we described in Sect. 4.

Let n, m, ℓ, e, κ be integers, $q = p^e$ be a modulus such that $m \geq 2n\log q$ and $\ell < n$, where p is a large prime and $p > \kappa$. Let χ be a noise distribution, and let $\sigma_x, \sigma_e, \gamma_x, \gamma_e > 0$ be parameters. The function evaluation sampling domain is $\mathsf{D}_\kappa^E = \mathsf{D}_x^E \times \mathsf{D}_e^E$, where D_x^E (resp. D_e^E) is the set of $\boldsymbol{x}$ (resp. $\boldsymbol{e}$) in $\mathbb{Z}^n$ (resp. $\mathbb{Z}^{2m}$) with $\|\boldsymbol{x}\| \leq \gamma_x\sqrt{n}\sigma_x$ (resp. $\|\boldsymbol{e}\| \leq \gamma_e\sqrt{2m}\sigma_e$). Its inversion domain is $\mathsf{D}_\kappa^D = \mathsf{D}_x^D \times \mathsf{D}_e^D$, where $\mathsf{D}_\mathsf{x}^\mathsf{D}$ (resp. D_e^D) is the set of $\boldsymbol{x}$ (resp. $\boldsymbol{e}$) in $\mathbb{Z}^n$ (resp. $\mathbb{Z}^{2n}$) with $\|\boldsymbol{x}\| \leq \sqrt{n}\sigma_x$ (resp. $\|\boldsymbol{e}\| \leq \sqrt{2m}\sigma_e$), and its range is $\mathsf{R} = \mathbb{Z}_q^{2m}$. In this case, the function inputs are sampled from the distribution $D_{\mathsf{D}_\kappa^E} = D_{\mathbb{Z}^n,\sigma_x}^{\mathsf{D}_x^E} \times D_{\mathbb{Z}^{2m},\sigma_e}^{\mathsf{D}_e^E}$. We remark that $D_{\mathbb{Z}^n,\sigma_x}^{\mathsf{D}_x^E}$ (resp. $D_{\mathbb{Z}^{2m},\sigma_e}^{\mathsf{D}_e^E}$) is obtained by restricting the distribution $D_{\mathbb{Z}^n,\sigma_x}$ (resp. $D_{\mathbb{Z}^{2m},\sigma_e}$) to the support of D_x^E (resp. D_e^E).

Furthermore, let $\mathsf{HE} = (\mathsf{HE.KeyGen}, \mathsf{HE.Enc}, \mathsf{HE.Dec}, \mathsf{HE.Eval})$ be a leveled fully homomorphic encryption scheme that can homomorphically evaluate PRF presented in Sect. 4 with polynomial modulus. Let $(\mathsf{Eval}^{\mathsf{Pub}}, \mathsf{Eval}^{\mathsf{Trap}})$ be a pair of deterministic algorithms that are δ-compatible for HE.Dec. Specifically, this δ might be $4^d m^{3/2}$ or $\tilde{O}(n^{2+\epsilon})$ according to different homomorphic evaluation algorithms according to Theorem 2.14. Furthermore, we use $k_3 \in \mathbb{N}$ to denote the output length of HE.Eval.

Construction. Below we present our construction of $\mathsf{ABM\text{-}LTF}$. Our scheme modifies that of Libert et al. [41] in an essential way. To highlight our new insights, we describe our modifications in the boxes.

- **Key generation.** $\mathsf{ABM.Gen}(1^\kappa)$ does the following steps:
  1. Compute and output $\bar{\mathbf{A}} = \mathbf{C} \cdot \mathbf{B} + \mathbf{F} \in \mathbb{Z}_q^{n \times m}$ with $\mathbf{B} \xleftarrow{\$} U(\mathbb{Z}_q^{\ell \times m})$, $\mathbf{C} \xleftarrow{\$} U(\mathbb{Z}_q^{n \times \ell})$ and $\mathbf{F} \leftarrow \chi^{n \times m}$.
  2. Select a secure pseudorandom function $\mathsf{PRF} : \{0,1\}^k \times \{0,1\}^v \rightarrow \{0,1\}^\kappa$ with input length $v \in \mathbb{N}$ and key length $k \in \mathbb{N}$. Choose $K \xleftarrow{\$} \{0,1\}^k$ as an independent key for PRF. We denote by $s_i \in \{0,1\}$ the i-th bit of K.
  3. Run $\mathsf{HE.KeyGen}$ algorithm of a HE scheme $\boxed{(\mathsf{hek}, \mathsf{hevk}, \mathsf{hdk}) \leftarrow \mathsf{HE.KeyGen}}$. Express the decryption algorithm $\mathsf{HE.Dec}$ as a NAND Boolean circuit $\boxed{C_{\mathsf{Dec}}}$, and express its decryption key hdk as $\boxed{\mathsf{hdk} = (hdk_1, ..., hdk_g)}$ where $hdk_i \in \{0,1\}$ and $g \in \mathbb{N}$.
  4. Select g low-norm matrices $\{R_{\mathbf{D}i}\}_{i \in [g]} \xleftarrow{\$} \{-1,1\}^{m \times m}$.
  5. Set $\boxed{\boldsymbol{c}_b \xleftarrow{\$} \mathsf{HE.Enc}(\mathsf{hek}, b)}$ for $b = 0, 1$.
  6. Set $\boxed{\boldsymbol{d}_i \xleftarrow{\$} \mathsf{HE.Enc}(\mathsf{hek}, s_i)}$ for $i \in [k]$.
  7. Set $\boxed{\mathbf{D}_i = \bar{\mathbf{A}} \cdot R_{\mathbf{D}_i} + hdk_i \mathbf{G}}$ for $i \in [g]$.
  8. Output the evaluation key ek, the inversion key ik and the lossy generation key tk, which consist of
$$\mathsf{ek} = \left(\mathsf{PRF}, C_{\mathsf{PRF}}, C_{\mathsf{Dec}}, \bar{\mathbf{A}}, \{\boldsymbol{d}_i\}_{i \in [k]}, \{\mathbf{D}_i\}_{i \in [g]}, \boldsymbol{c}_0, \boldsymbol{c}_1, \mathsf{hevk}\right),$$
$$\mathsf{ik} = \left(\{\mathbf{R}_{\mathbf{D}_i}\}_{i \in [g]}, \mathsf{hdk}, K\right), \qquad \mathsf{tk} := K.$$
- **Evaluation.** $\mathsf{ABM.Eval}(\mathsf{ek}, \mathsf{t}, \mathsf{X})$ takes as inputs $\mathsf{X} := (\boldsymbol{x}, \boldsymbol{e}) \in \mathsf{D}_\kappa^E$ and the tag $\mathsf{t} = (\mathsf{t_c}, \mathsf{t_a}) \in \{0,1\}^\kappa \times \{0,1\}^v$, and proceeds as follows.
  1. For each integer $j \in [\kappa]$, let $C_{\mathsf{PRF},j} : \{0,1\}^k \times \{0,1\}^v \rightarrow \{0,1\}$ be the Boolean circuit, which evaluate the j-th bit of $\mathsf{PRF}(K, \mathsf{t_a}) \in \{0,1\}^\kappa$. Run the homomorphic evaluation algorithm of HE to obtain
$$\boxed{\mathsf{ct}_j = \mathsf{HE.Eval}(\mathsf{hevk}, C_{\mathsf{PRF},j}, (\{\boldsymbol{d}_i\}_{i \in [k]}, \{\boldsymbol{c}_{\mathsf{t_a}[i]}\}_{i \in [\ell]}))},$$
where $\mathsf{t_a}[i]$ denotes the i-th bit of $\mathsf{t_a}$ for $i \in [\ell]$. Furthermore, run the public evaluation algorithm to obtain
$$\boxed{\mathbf{B}_{\mathsf{PRF},j} = \mathsf{Eval}^{\mathsf{Pub}}(C_{\mathsf{Dec}}, (\{\mathbf{D}_i\}_{i \in [g]}, \{(\mathsf{ct}_j)_i \mathbf{G}\}))},$$
where $\{(\mathsf{ct}_j)_i\}_{i \in \mathbb{N}}$ denotes the bit representation of ciphertext ct_j.
  2. Define the matrix
$$\mathbf{A}_\mathsf{t} = \left(\bar{\mathbf{A}}, \textstyle\sum_{j \in [\kappa]} \left((-1)^{\mathsf{t_c}[j]} \mathbf{B}_{\mathsf{PRF},j} + \mathsf{t_c}[j]\mathbf{G}\right)\right) \in \mathbb{Z}_q^{n \times 2m},$$
and compute the output $\boldsymbol{y}^t = \boldsymbol{x}^t \cdot \mathbf{A}_\mathsf{t} + \boldsymbol{e}^t \in \mathbb{Z}_q^{2m}$. Notice that after summation for all $j \in [\kappa]$, the coefficient of matrix $\mathbf{G}$ in the right half part of $\mathbf{A}_\mathsf{t}$ is just the hamming distance between $\mathsf{t_c}$ and $\mathsf{PRF}(K, \mathsf{t_a})$.

- **Inversion.** $\mathsf{ABM.Invert}(\mathsf{ik}, \mathsf{t}, \mathsf{Y})$ takes as inputs the inversion key $\mathsf{ik} = \left(\{\mathbf{R}_{\mathbf{D}_i}\}_{i\in[g]}, K\right)$, the tag $\mathsf{t} = (\mathsf{t_c}, \mathsf{t_a}) \in \{0,1\}^\kappa \times \{0,1\}^\ell$ and $\mathsf{Y} := \boldsymbol{y} \in \mathsf{R}$, and proceeds:
  1. Return $\perp$ if $\mathsf{t_c} = \mathsf{PRF}(K, \mathsf{t_a})$.
  2. Otherwise, for each $j \in [\kappa]$, run the following two algorithms:

     $$\boxed{\mathsf{ct_{id}} = \mathsf{HE.Eval}(\mathsf{hevk}, C_{\mathsf{PRF},j}, (\{\boldsymbol{d}_i\}_{i\in[k]}, \{\boldsymbol{c}_{\mathsf{t_a}[i]}\}_{i\in[\ell]}))}$$

     $$\boxed{\mathbf{R}_{\mathsf{PRF},j} = \mathsf{Eval}^{\mathsf{Trap}}(C_{\mathsf{Dec}}, \bar{\mathbf{A}}, (\{hdk_i\}_{i\in[g]}, \{(\mathsf{ct_{id}})_i\}_{i\in[k_3]}), \{\mathbf{R}_{\mathbf{D}_i}\}_{i\in[g]}, \{[0]_i\}_{i\in[k_3]})}$$

     and compute the matrix $\mathbf{R}_\mathsf{t} = \sum_{j\in[\kappa]}(-1)^{\mathsf{t_c}[j]}\mathbf{R}_{\mathsf{PRF},j} \in \mathbb{Z}^{m\times m}$, where for each $i \in [k_3]$, $[0]_i$ denotes 0 matrix with dimension $m \times m$.
  3. Let h_t denote the hamming distance between $\mathsf{t_c}$ and PRF $(K, \mathsf{t_a})$. Then Compute and set $\mathbf{A}_\mathsf{t} = \left(\bar{\mathbf{A}}, \bar{\mathbf{A}}\mathbf{R}_\mathsf{t} + h_\mathsf{t}\mathbf{G}\right) \in \mathbb{Z}_q^{n\times 2m}$, Use the $\mathbf{G}$-trapdoor $\mathbf{R}_\mathsf{t}$ of $\mathbf{A}$ with tag h_t to solve the unique $(\boldsymbol{x}, \boldsymbol{e}) \in \mathsf{D}_\kappa^D$ such that $\boldsymbol{y}^t = \boldsymbol{x}^t \cdot \mathbf{A} + \boldsymbol{e}^t$. This can be done by applying the LWE inversion algorithm (which can be found in full version of this paper).
- **Lossy tag generation.** $\mathsf{ABM.LTag}(\mathsf{tk})$ takes as input an auxiliary tag component $\mathsf{t_a} \in \{0,1\}^\ell$ and uses $\mathsf{tk} = K$ to compute and output $\mathsf{t_c} = \mathsf{PRF}(K, \mathsf{t_a}) \in \{0,1\}^\kappa$.

Below we state a theorem that summarizes what we can achieve. Due to space limit, we present the syntax of ABM-LTF and the security analysis in full version of this paper.

Theorem 5.1. *Let κ be the security parameter, $\chi = D_{\mathbb{Z},\beta/(2\sqrt{\kappa})}$ for some $\beta > 4\kappa$. Let n, m, ℓ, e be functions of κ, $q = p^e$ be a modulus such that $m \geq 2n\log q$, $n = \Omega(\ell \log q)$ and $\kappa < \ell < n$, where p is a large prime and $p > \kappa$. Let $\gamma_x \geq 3\sqrt{m/n}$, $\gamma_e \geq 3$, $\sigma_x > \Omega(n)$, $\Omega(m\sqrt{n}\kappa\delta\beta\sigma_x) \leq \sigma_e \leq q/(10\sqrt{2}\kappa\delta m)$. Then, our new construction is an l-lossy* ABM-LTF *with $l = \Omega(n\log n)$ based on* $\mathsf{LWE}_{\ell,2m,q,\chi}$.

Acknowledgements. We would like to thank Bo Yang and Mingsheng Wang for their helpful discussions. We also thank the anonymous reviewers of PKC 2020 for their insightful advices. Qiqi Lai is supported by the National Key R&D Program of China (2017YFB0802000), the National Natural Science Foundation of China (61802241, 61772326, 61802242, 61802075), the Natural Science Basic Research Plan in Shaanxi Province of China (2019JQ-360) and the National Cryptography Development Foundation during the 13th Five-year Plan Period (MMJJ20180217). Feng-Hao Liu is supported by the NSF Award CNS-1657040. Zhedong Wang is supported by the NSF Award CNS-1657040 and the National Key R&D Program of China-2017YFB0802202. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsors.

References

1. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_28
2. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: 28th ACM STOC, pp. 99–108. ACM Press, May 1996
3. Alperin-Sheriff, J., Peikert, C.: Faster bootstrapping with polynomial error. In: Garay and Gennaro [27], pp. 297–314
4. Alwen, J., Krenn, S., Pietrzak, K., Wichs, D.: Learning with rounding, revisited - new reduction, properties and applications. In: Canetti and Garay [23], pp. 57–74
5. Attrapadung, N., Hanaoka, G., Yamada, S.: A framework for identity-based encryption with almost tight security. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9452, pp. 521–549. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48797-6_22
6. Bai, S., Langlois, A., Lepoint, T., Stehlé, D., Steinfeld, R.: Improved security proofs in lattice-based cryptography: using the Rényi divergence rather than the statistical distance. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015, Part I. LNCS, vol. 9452, pp. 3–24. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48797-6_1
7. Banerjee, A., Peikert, C.: New and improved key-homomorphic pseudorandom functions. In: Garay and Gennaro [27], pp. 353–370
8. Banerjee, A., Peikert, C., Rosen, A.: Pseudorandom functions and lattices. In: Pointcheval and Johansson [48], pp. 719–737
9. Bellare, M., Rogaway, P.: The exact security of digital signatures-how to sign with RSA and Rabin. In: Maurer, U. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 399–416. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-68339-9_34
10. Blazy, O., Kiltz, E., Pan, J.: (Hierarchical) identity-based encryption from affine message authentication. In: Garay and Gennaro [27], pp. 408–425
11. Bogdanov, A., Guo, S., Masny, D., Richelson, S., Rosen, A.: On the hardness of learning with rounding over small modulus. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016-A, Part I. LNCS, vol. 9562, pp. 209–224. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49096-9_9
12. Bogdanov, A., Rosen, A.: Pseudorandom functions: three decades later. Cryptology ePrint Archive, Report 2017/652 (2017). https://eprint.iacr.org/2017/652
13. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_13
14. Boneh, D., et al.: Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 533–556. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55220-5_30
15. Boneh, D., Lewi, K., Montgomery, H.W., Raghunathan, A.: Key homomorphic PRFs and their applications. In: Canetti and Garay [23], pp. 410–428
16. Boyen, X., Li, Q.: Towards tightly secure lattice short signature and id-based encryption. In: Cheon and Takagi [25], pp. 404–434
17. Boyen, X., Li, Q.: All-but-many lossy trapdoor functions from lattices and applications. In: Katz and Shacham [38], pp. 298–331
18. Brakerski, Z.: Fully homomorphic encryption without modulus switching from classical GapSVP. In: Safavi-Naini and Canetti [50], pp. 868–886

19. Brakerski, Z., Goldwasser, S.: Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: quadratic residuosity strikes back). In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 1–20. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14623-7_1
20. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: Boneh, D., Roughgarden, T., Feigenbaum, J. (eds.) 45th ACM STOC, pp. 575–584. ACM Press, June 2013
21. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. In: Ostrovsky, R. (ed.) 52nd FOCS, pp. 97–106. IEEE Computer Society Press (2011)
22. Brakerski, Z., Vaikuntanathan, V.: Lattice-based FHE as secure as PKE. In: Naor, M. (ed.) ITCS 2014, pp. 1–12. ACM, January 2014
23. Canetti, R., Garay, J.A. (eds.): CRYPTO 2013, Part I. LNCS, vol. 8042. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40041-4
24. Chen, J., Wee, H.: Fully, (almost) tightly secure IBE and dual system groups. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 435–460. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40084-1_25
25. Cheon, J.H., Takagi, T. (eds.): ASIACRYPT 2016, Part II. LNCS, vol. 10032. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53890-6
26. Döttling, N., Schröder, D.: Efficient pseudorandom functions via on-the-fly adaptation. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015, Part I. LNCS, vol. 9215, pp. 329–350. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-47989-6_16
27. Garay, J.A., Gennaro, R. (eds.): CRYPTO 2014, Part I. LNCS, vol. 8616. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44371-2
28. Gay, R., Hofheinz, D., Kiltz, E., Wee, H.: Tightly CCA-Secure Encryption Without Pairings. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016, Part I. LNCS, vol. 9665, pp. 1–27. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49890-3_1
29. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Mitzenmacher [44], pp. 169–178
30. Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based. In: Canetti and Garay [23], pp. 75–92
31. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions (extended abstract). In: 25th FOCS, pp. 464–479. IEEE Computer Society Press, October 1984
32. Gong, J., Chen, J., Dong, X., Cao, Z., Tang, S.: Extended nested dual system groups, revisited. In: Cheng, C.-M., Chung, K.-M., Persiano, G., Yang, B.-Y. (eds.) PKC 2016, Part I. LNCS, vol. 9614, pp. 133–163. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49384-7_6
33. Hofheinz, D., Jager, T.: Tightly secure signatures and public-key encryption. In: Safavi-Naini and Canetti [50], pp. 590–607
34. Jager, T., Kurek, R., Pan, J.: Simple and more efficient PRFs with tight security from LWE and matrix-DDH. Cryptology ePrint Archive, Report 2018/826, to be appear in Asiacrypt 2018. https://eprint.iacr.org/2018/826
35. Joye, M., Libert, B.: Efficient cryptosystems from 2^k-th power residue symbols. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 76–92. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38348-9_5

36. Katsumata, S., Yamada, S.: Partitioning via non-linear polynomial functions: more compact IBEs from ideal lattices and bilinear maps. In: Cheon and Takagi [25], pp. 682–712
37. Katz, J., Lindell, Y.: Introduction to Modern Cryptography, 2nd edn. CRC Press, Boca Raton (2014)
38. Katz, J., Shacham, H. (eds.): CRYPTO 2017, Part III. LNCS, vol. 10403. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63697-9
39. Katz, J., Wang, N.: Efficiency improvements for signature schemes with tight security reductions. In: Jajodia, S., Atluri, V., Jaeger, T. (eds.) ACM CCS 2003, pp. 155–164. ACM Press, New York (2003)
40. Lewko, A.B., Waters, B.: Efficient pseudorandom functions from the decisional linear assumption and weaker variants. In: Al-Shaer, E., Jha, S., Keromytis, A.D. (eds.) ACM CCS 2009, pp. 112–120. ACM Press, New York (2009)
41. Libert, B., Sakzad, A., Stehlé, D., Steinfeld, R.: All-but-many lossy trapdoor functions and selective opening chosen-ciphertext security from LWE. In: Katz and Shacham [38], pp. 332–364
42. Micciancio, D., Mol, P.: Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 465–484. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_26
43. Micciancio, D., Peikert, C.: Trapdoors for lattices: simpler, tighter, faster, smaller. In: Pointcheval and Johansson [48], pp. 700–718
44. Mitzenmacher, M. (ed.): 41st ACM STOC. ACM Press, May/June 2009
45. Naor, M., Reingold, O.: Number-theoretic constructions of efficient pseudo-random functions. In: 38th FOCS, pp. 458–467. IEEE Computer Society Press, October 1997
46. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_16
47. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In Mitzenmacher [44], pp. 333–342
48. Pointcheval, D., Johansson, T. (eds.): EUROCRYPT 2012. LNCS, vol. 7237. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4
49. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) 37th ACM STOC, pp. 84–93. ACM Press, May 2005
50. Safavi-Naini, R., Canetti, R. (eds.): CRYPTO 2012. LNCS, vol. 7417. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5
51. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985). https://doi.org/10.1007/3-540-39568-7_5
52. Vadhan, S.P.: Pseudorandomness. Found. Trends Theor. Comput. Sci. **7**(1–3), 1–336 (2012)
53. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_7
54. Yamada, S.: Asymptotically compact adaptively secure lattice IBEs and verifiable random functions via generalized partitioning techniques. In: Katz and Shacham [38], pp. 161–193