Deterministic Identity-Based Encryption from Lattice-Based Programmable Hash Functions with High Min-Entropy

Daode Zhang[1,2,3], Jie Li[1,2,3], Bao Li[1,2,3], Xianhui Lu[1,2,3], Haiyang Xue[1,2,3], Dingding Jia[1,2,3] and Yamin Liu[1,2,3]

1 School of Cyber Security, University of Chinese Academy of Sciences.
2 State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China.
3 Data Assurances and communications Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China.
zhangdaode0119@gmail.com

Abstract. There only exists one deterministic identity-based encryption (DIBE) scheme which is adaptively secure in the auxiliary-input setting, under the learning with errors (LWE) assumption. However, the master public key consists of $\mathcal{O}(\lambda)$ basic matrices. In this paper, we consider to construct adaptively secure DIBE schemes with more compact public parameters from the LWE problem.

- On the one hand, we gave a generic DIBE construction from lattice-based programmable hash functions with high min-entropy.
- On the other hand, when instantiating our generic DIBE construction with four LPHFs with high min-entropy, we can get four adaptively secure DIBE schemes with more compact public parameters. In one of our DIBE schemes, the master public key only consists of $\omega(\log \lambda)$ basic matrices.

Keywords: deterministic identity-based encryption, adaptively secure, auxiliary-input, compact public parameters, the learning with errors, lattice-based programmable hash functions with high min-entropy.

1 Introduction

A DIBE scheme is an identity-based encryption (IBE) scheme [17] whose encryption algorithm is deterministic. This primitive was proposed by Bellare et al. [5] via extending the security definition under high min-entropy into the identity-based setting. In order to construct DIBE schemes, Bellare et al. [5] first defined a notion of identity-based lossy trapdoor functions (IB-LTDFs). And they obtained a DIBE scheme by constructing an IB-LTDF with a universal property, based on the DLIN assumption. However, due to the inherent limitation of IB-LTDFs, their scheme can only achieve a selective security, i.e., the adversary must commit an challenge identity before getting the master public key from the challenger.

In SCN12, Xie et al. [18] gave a more efficient secure DIBE scheme in the auxiliary-input setting, based on the hardness of the LWE problem. In their scheme, there exists only 3 matrices in the master public key. However, the scheme only satisfies a selective security as same as the scheme in [5]. The more significant contribution of Xie et al. [18] is that they proposed the first DIBE scheme with a much more realistic adaptive security (or equivalently, full security) in the auxiliary-input setting, based on the same assumption. To our best knowledge, their scheme is the only DIBE scheme that achieves an adaptive security. However, their scheme requires $\ell + 2$ basic matrices in the master public key so that it is less efficient than their selectively secure scheme, where ℓ is the bit length of the identity and $\ell = \Theta(\lambda)$.

Fig. 1. Comparison of Adaptively Secure DIBE Schemes in the Auxiliary-Input Setting.

Schemes	# of $\mathbb{Z}_q^{n\times m}$ matrix |mpk|	Rounding Parameter p	Message Space t	Sample Width σ	Reduction Cost
XXZ12 [18]	$\mathcal{O}(\lambda)$	$\widetilde{\mathcal{O}}(n^{4.5+3\eta})$	$\widetilde{\mathcal{O}}(n^{3.5+2\eta})$	$\widetilde{\mathcal{O}}(n^{2.5+\eta})$	$\mathcal{O}(\frac{\epsilon}{\ell Q})$
Ours:					
$\mathrm{DIBE}_{\mathrm{MAH}}$	$\omega(\log^2 \lambda)$	$\widetilde{\mathcal{O}}(n^{6.5+5.5\eta})$	$\widetilde{\mathcal{O}}(n^{6+5\eta})$	$\widetilde{\mathcal{O}}(n^{5+4\eta})$	$\mathcal{O}(\frac{\epsilon^{\varphi+1}}{Q^{\varphi}})^{\dagger}$
$\mathrm{DIBE}_{\mathrm{AFF}}$	$\omega(\log \lambda)$	$\mathsf{poly}(n)$	$\mathsf{poly}(n)$	$\mathsf{poly}(n)$	$\mathcal{O}(\frac{\epsilon^2}{\ell^2 Q})$
$\mathrm{DIBE}_{\mathrm{Yam16}}$	$\sqrt{\lambda}$	$\widetilde{\mathcal{O}}(n^{c+3.5+2.5\eta})$	$\widetilde{\mathcal{O}}(n^{c+3+2\eta})$	$\widetilde{\mathcal{O}}(n^{c+2+\eta})^{\S}$	$\mathcal{O}(\frac{\epsilon^3}{\ell Q^2})$
$\mathrm{DIBE}_{\mathrm{ZCZ16}}$	$\mathcal{O}(\log Q)$	$\widetilde{\mathcal{O}}(n^{c+4+3\eta})$	$\widetilde{\mathcal{O}}(n^{c+3.5+2.5\eta})$	$\widetilde{\mathcal{O}}(n^{c+2.5+1.5\eta})^{\ddagger}$	$\mathcal{O}(\frac{\epsilon}{\ell Q^2})$

|mpk| shows the size of the master public key. Q and ϵ denote the number of key extraction queries and the advantage of the adversary, respectively. $\mathsf{poly}(n)$ represents a fixed but large polynomial that does not depend Q and ϵ. To measure the reduction cost, we show the advantage of the LWE algorithm constructed from the adversary against the corresponding DIBE scheme.

†, $\varphi > 1$ is the constant satisfying $s = 1 - 2^{-\frac{1}{\varphi}}$, where $s \in \{0, 1\}$ is the relative distance of the underlying error correcting code. We can take φ as close to 1 as one wants.

§, $c = c_1 + c_2$ and c_1, c_2 are the smallest integers satisfying that $\frac{n^{c_1}}{2} \geq Q + 1$ and $n^{-c_2} \leq \epsilon$.

‡, c is the smallest integer satisfying that $n^c \geq Q + 1$.

Our Contributions. In this paper, we consider to construct adaptively secure DIBE schemes with more compact public parameters from the LWE problem.

- We gave a generic DIBE construction from lattice-based programmable hash functions (LPHFs) with high min-entropy [22]. Note that the adaptively secure DIBE in [18] is in our framework.
- We present more instantiations of LPHFs with high min-entropy. In fact, most of these instantiations are already implicit in recent works. Following the works of Zhang et al. [22] who proved that the IBE schemes in [1, 15, 22] implies instantiations of LPHFs with high min-entropy, we show that

LPHFs with high min-entropy can be constructed from partitioning functions with compatible algorithms [20]. And we show that the IBE schemes in [19, 14, 4] naturally imply instantiations of LPHFs with high min-entropy. Combining with the result of Zhang et al., we conclude that the adaptively secure and anonymous IBE schemes in [1, 15, 22, 4, 19, 14, 20] naturally imply instantiations of LPHFs with high min-entropy [1].

- When instantiating our generic DIBE construction with four LPHFs with high min-entropy in [20, 19, 22], we can get four adaptively secure DIBE schemes with more compact public parameters. In our DIBE schemes, the master public key respectively consists of $\omega(\log^2 \lambda), \omega(\log \lambda), \sqrt{\lambda}, \mathcal{O}(\log Q)$ number of basic matrices, where Q denotes the number of key extraction queries. Please see more details in Figure 1.

Related Works. In [5], Bellare et al. extended the notion of lossy trapdoor function (LDTF) to identity-setting and introduced the notion of identity-based LTDF (IB-LTDF). And they used IB-LTDF to construct DIBE scheme with a selective security from pairings. Soon afterwards, Escala et al. [10] extended the notion of IB-LTDF [5] and introduced the notion of hierarchical identity-based trapdoor functions (HIB-TDFs). With HIB-TDFs, they could construct deterministic hierarchical identity-based schemes (DHIBE). They instantiated HIB-TDFs from pairings so that they constructed a pairing-based DHIBE scheme. Fang et al. [11] constructed a DHIBE scheme with a selective security based on the hardness of the learning with rounding problem over small modulus [6]. In fact, a DHIBE with a selective security implies a selectively secure DIBE. In SCN12, Xie et al. [18] gave a more efficient DIBE scheme with a security security. Additionally, they also proposed the first and the only DIBE scheme with an adaptive security in the auxiliary-input setting.

Remarks. This work is very relevant to [21] in which we constructed the DIBE schemes $\mathrm{DIBE}_{\mathrm{MAH}}$, $\mathrm{DIBE}_{\mathrm{AFF}}$ and $\mathrm{DIBE}_{\mathrm{Yam16}}$ directly from the works of Yamada [19, 20]. As our growing understanding, we find that all adaptively secure DIBE schemes in [18, 21] can be explained by using LPHFs with high min-entropy [2]. So, in this paper, we present a generic DIBE construction from LPHFs with high min-entropy.

2 Preliminaries

Notations. Let λ be the security parameter, and all other quantities are implicitly dependent on λ. Let $\mathsf{negl}(\lambda)$ denote a negligible function and $\mathsf{poly}(\lambda)$ denote an unspecified function $f(\lambda) = \mathcal{O}(\lambda^c)$ for some constant c. A function f is ϵ-hard-to-invert with respect to the distribution $\mathcal{D}$, if given $h(x)$ with $x \xleftarrow{\$} \mathcal{D}$,

[1] Note that Boyen and Li [7] constructed an adaptively secure and anonymous IBE scheme with tight security. However, their construction does not imply a LPHF and is not in our framework.

[2] Note that the adaptively secure DIBE scheme in [18] is constructed from the LPHF with high min-entropy in [1, 15].

there exists no PPT algorithm can find x with probability better than ϵ. For $n \in \mathbb{N}$, we use $[n]$ to denote a set $\{1, \cdots, n\}$. And for integer $q \geq 2$, $\mathbb{Z}_q$ denotes the quotient ring of integer modulo q. We use bold capital letters to denote matrices, such as $\mathbf{A}, \mathbf{B}$, and bold lowercase letters to denote column vectors, such as $\mathbf{x}, \mathbf{y}$. The notations $\mathbf{A}^\top$ and $[\mathbf{A}|\mathbf{B}]$ denote the transpose of the matrix $\mathbf{A}$ and the matrix of concatenating $\mathbf{A}$ and $\mathbf{B}$, respectively.

For $n \in \mathbb{N}$, we use $[n]$ to denote a set $\{1, \cdots, n\}$. For integer $q \geq 2$, $\mathbb{Z}_q$ denotes the quotient ring of integer modulo q. For integers $q \geq p \geq 2$ and $x \in \mathbb{Z}_q$, a rounding function $\lfloor \cdot \rceil_p : \mathbb{Z}_q \to \mathbb{Z}_p$ is defined by $\lfloor x \rceil_p = \lfloor (p/q) \cdot x \rceil \bmod p$.

2.1 Deterministic Identity-Based Encryption and Its Security

A deterministic identity-based encryption scheme DIBE with the identity space ID can be defined by a tuple of PPT algorithms $\mathsf{DIBE.Setup}, \mathsf{DIBE.KGen}, \mathsf{DIBE.Enc}, \mathsf{DIBE.Dec}$. The $\mathsf{DIBE.Setup}$ algorithm takes a security parameter 1^λ as input and outputs a master secret key mpk and a master secret key $msk \in \mathcal{M}$. The $\mathsf{DIBE.KGen}$ algorithm takes $mpk, msk, id \in \mathrm{ID}$ as input and outputs a private key sk_{id}. The deterministic algorithm $\mathsf{DIBE.Enc}$ takes $mpk, id \in \mathrm{ID}$ and a message msg, outputs a ciphertext c. The deterministic algorithm $\mathsf{DIBE.Dec}$ decrypts ciphertexts using the private key sk_{id}. We require that for all λ, all $id \in \mathrm{ID}$, and all $msg \in \mathcal{M}$, $\Pr[\mathsf{DIBE.Dec}(mpk, sk_{id}, id, \mathsf{DIBE.Enc}(mpk, id, msg)) = msg] = 1 - \mathsf{negl}(\lambda)$.

Definition 1 ([18]). *We say that a DIBE scheme* DIBE *is* PRIV1-ID-INDr*-secure with respect to ϵ-hard-to-invert auxiliary inputs if for any PPT algorithm $\mathcal{A}$, for any efficiently sampled distribution $\widehat{\mathcal{M}}$, and any efficiently computable $\mathcal{H}_{\mathrm{hard}} = \{h\}$ that is ϵ-hard-to-invert with respect to $\widehat{\mathcal{M}}$, such that the advantage of $\mathcal{A}$ in the following game is negligible.*

Setup. *At the outset of the game, the challenger runs $\mathsf{DIBE.Setup}(1^\lambda)$ which outputs a pair (mpk, msk) and gives mpk to $\mathcal{A}$.*

Phase 1. *When $\mathcal{A}$ adaptively makes key-extraction queries to the challenger, the challenger returns $sk_{id} \leftarrow \mathsf{DIBE.KGen}(mpk, msk, id)$, for all id in the key-extraction queries.*

Challenge Phase. *At some point, $\mathcal{A}$ outputs an identity id^*, on which it wishes to be challenged. Then, the challenger picks a random coin $\mathsf{coin} \xleftarrow{\$} \{0,1\}$, a message $msg \xleftarrow{\$} \widehat{\mathcal{M}}$, a random ciphertext c_1^* from the ciphertext space $\mathcal{C}$ and a function $h \xleftarrow{\$} \mathcal{H}_{\mathrm{hard}}$. If $\mathsf{coin} = 0$, it runs $\mathsf{DIBE.Enc}(mpk, id^*, msg) \to c_0^*$ and gives the challenge ciphertext $(c_0^*, h(msg))$ to $\mathcal{A}$. If $\mathsf{coin} = 1$, it gives $(c_1^*, h(msg))$ to $\mathcal{A}$.*

Phase 2. *$\mathcal{A}$ can also adaptively make key-extraction queries to the challenger, with the restriction $id \neq id^*$.*

Gauss. *Finally, $\mathcal{A}$ makes a guess coin' for coin.*

The advantage of $\mathcal{A}$ is defined as $\Pr\left[\mathsf{coin}' = \mathsf{coin}\right] - \frac{1}{2}$.

2.2 LPHFs with High Min-Entropy [22]

Let $\ell, \overline{m}, m, n, q, v$ be some polynomials in the security parameter λ. By $\mathcal{I}_n$ we denote the set of invertible matrices in $\mathbb{Z}_q^{n\times n}$. A hash function $\mathcal{H} : \mathrm{ID} \to \mathbb{Z}_q^{n\times m}$ consists of two algorithms $(\mathcal{H}.\mathsf{Gen}, \mathcal{H}.\mathsf{Eval})$. Given the security parameter λ, the probabilistic polynomial time (PPT) key generation algorithm $\mathcal{H}.\mathsf{Gen}(1^\lambda)$ outputs a key K, i.e., $K \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda)$. For any input $id \in \mathrm{ID} = \{0,1\}^\ell$, the efficiently deterministic evaluation algorithm $\mathcal{H}.\mathsf{Eval}(K, id)$ outputs a hash value $\mathbf{Z} \in \mathbb{Z}_q^{n\times m}$, i.e., $\mathbf{Z} = \mathcal{H}.\mathsf{Eval}(K, id)$.

Definition 2 (LPHFs). *A hash function $\mathcal{H} : \mathrm{ID} \to \mathbb{Z}_q^{n\times m}$ is a $(1, v, \beta, \gamma, \delta)$-LPHF if there exist a PPT trapdoor key generation algorithm $\mathcal{H}.\mathsf{TrapGen}$ and a PPT deterministic trapdoor evaluation algorithm $\mathcal{H}.\mathsf{TrapEval}$ such that given a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times\overline{m}}$ and a (public) trapdoor matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$ the following properties hold:*

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

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

$$\mathcal{H}.\mathsf{Eval}(K', id) = \mathbf{A}\mathbf{R}'_{id} + \mathbf{S}'_{id}\mathbf{B}.$$

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

Well-distributed hidden matrices : *For all $(K', td) \leftarrow \mathcal{H}.\mathsf{TrapGen}(1^\lambda, \mathbf{A}, \mathbf{B})$, any inputs $id^*, id_1, \cdots, id_v$ such that $id^* \neq id_j$ for any $j \in [v]$, we have that*

$$\Pr[\mathbf{S}'_{id^*} = \mathbf{0} \wedge \mathbf{S}'_{id_1}, \cdots, \mathbf{S}'_{id_v} \in \mathcal{I}_n] \ge \delta,$$

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

Definition 3 (LPHFs with High Min-Entropy). *Let $\mathcal{H} : \mathrm{ID} \to \mathbb{Z}_q^{n\times m}$ be a $(1, v, \beta, \gamma, \delta)$-LPHF with $\gamma = \mathsf{negl}(\lambda)$ and noticeable $\delta > 0$. Let K be the key space of $\mathcal{H}$, and let $\mathcal{H}.\mathsf{TrapGen}$ and $\mathcal{H}.\mathsf{TrapEval}$ be a pair of trapdoor generation and trapdoor evaluation algorithms for $\mathcal{H}$. We say that $\mathcal{H}$ is a LPHF with high min-entropy if for uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times\overline{m}}$ and a (public) trapdoor matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$, the following condition holds*

- Property 1. *For any $(K', td) \leftarrow \mathcal{H}.\mathsf{TrapGen}(1^\lambda, \mathbf{A}, \mathbf{B})$, any $id \in \mathrm{ID}$ and its corresponding $(\mathbf{R}'_{id}, \mathbf{S}'_{id}) = \mathcal{H}.\mathsf{TrapEval}(td, K', id)$, the statistical distance between $(\mathbf{A}, K', \mathbf{v}, \mathbf{u})$ and $(\mathbf{A}, K', \mathbf{v}, (\mathbf{R}'_{id})^\top \mathbf{v})$ is negligible in λ, where $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m, \mathbf{v} \xleftarrow{\$} \mathbb{Z}_q^{\overline{m}}$.* □

Remark 1. Note that this definition of LPHFs with min high-entropy is much weaker that Zhang et al's definition of LPHFs with min high-entropy which includes another one requirement. In [14], Katsumata and Yamada found that this requirement is not necessary, i.e., we can define this weaker version of LPHFs with min high-entropy while keeping their functionality–constructing IBE schemes.

3 Generic DIBE Construction

Here, we construct an adaptively secure DIBE scheme in the auxiliary-input setting by using a $(1, v, \beta, \gamma, \delta)$ LPHF $\mathcal{H}$ with high min-entropy from $\{0,1\}^\ell$ to $\mathbb{Z}_q^{n\times m}$, where γ is negligible and $\delta > 0$ is noticeable. Let $\mathcal{H}.\mathsf{TrapGen}$ and $\mathcal{H}.\mathsf{TrapEval}$ be a pair of trapdoor generation and trapdoor evaluation algorithm of $\mathcal{H}$ that satisfies the condition in Definition 3, where integers n, m, q, v, β are polynomials in the security parameter λ. Additionally, let integers $\overline{m} = \mathcal{O}(n \log q), m' = \overline{m} + m$. We assume ID $= \{0,1\}^\ell$ and $\mathcal{M} = \mathbb{Z}_t^n$, where ID is the user identity space and $\mathcal{M}$ is the message space. Our generic DIBE scheme $\mathsf{DIBE} = (\mathsf{DIBE.Setup}, \mathsf{DIBE.KGen}, \mathsf{DIBE.Enc}, \mathsf{DIBE.Dec})$ is defined as follows.

- **Setup**. Algorithm $\mathsf{DIBE.Setup}$ takes 1^λ as input, and generates a pair $(\mathbf{A}, \mathbf{T_A}) \xleftarrow{\$} \mathsf{TrapGen}(1^n, 1^{\overline{m}}, q)$, where $\mathbf{A} \in \mathbb{Z}_q^{n\times\overline{m}}$ and $\mathbf{T_A} \in \mathbb{Z}_q^{\overline{m}\times\overline{m}}$. Then, it obtains $K \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda)$. Finally, it outputs
$$mpk = (\mathbf{A}, K) \qquad \text{and} \qquad msk = \mathbf{T_A}.$$
- **Key Generation**. Algorithm $\mathsf{DIBE.KGen}$ takes mpk and $id \in$ ID as inputs. It first computes $\mathbf{F}_{id} = [\mathbf{A}|\mathcal{H}.\mathsf{Eval}(K, id)] \in \mathbb{Z}_q^{n\times m'}$. Then generates $\mathbf{T}_{\mathbf{F}_{id}} \in \mathbb{Z}^{m'\times m'}$ by running $\mathsf{SampleBasisLeft}(\mathbf{A}, \mathcal{H}.\mathsf{Eval}(K, id), \mathbf{T_A}, \sigma)$. It finally outputs $sk_{id} = \mathbf{T}_{\mathbf{F}_{id}}$.
- **Encryption**. Algorithm $\mathsf{DIBE.Enc}$ takes $mpk, id \in$ ID, $\mathbf{m} \in \mathcal{M}$ as inputs. It first computers $\mathbf{F}_{id} = [\mathbf{A}|\mathcal{H}.\mathsf{Eval}(K, id)] \in \mathbb{Z}_q^{n\times m'}$. Then, it outputs the ciphertext $\mathbf{c} = \lfloor \mathbf{F}_{id}^\top \mathbf{m} \rceil_p$.
- **Decryption**. To decrypt a ciphertext $\mathbf{c}$ with a private key $sk_{id} = \mathbf{T}_{\mathbf{F}_{id}}$, the algorithm $\mathsf{DIBE.Dec}$ computers $\mathbf{m} \xleftarrow{\$} \mathsf{Invert}(\mathbf{c}, \mathbf{F}_{id}, sk_{id})$. Then, if $\mathbf{m} \in \mathbb{Z}_t^n$ it outputs $\mathbf{m}$, and otherwise it outputs $\perp$.

3.1 Correctness and Parameter Selection

In order to make sure the correctness of the DIBE scheme and make the security proof follow through, we need the following to satisfy.

- $\mathsf{TrapGen}$ in Lemma 4 (Item 1) can work ($\overline{m} \geq 6n\lceil \log q \rceil$), and it returns $\mathbf{T_A}$ satisfying $\|\widetilde{\mathbf{T_A}}\| \geq \mathcal{O}(\sqrt{n \log q})$.
- $\mathsf{SampleBasisLeft}$ in Lemma 4 (Item 2) can operate ($\sigma \geq \|\widetilde{\mathbf{T_A}}\| \cdot \omega(\sqrt{\log(\overline{m} + m)}) = \mathcal{O}(\sqrt{n \log q}) \cdot \omega(\sqrt{\log(\overline{m} + m)})$).

- $\mathsf{SampleBasisRight}$ in Lemma 4 (Item 3) can operate ($\sigma \geq s_1(\mathbf{R}_{id}) \cdot \omega(\sqrt{\log m}) \geq \beta \cdot \omega(\sqrt{\log m})$).
- In order to keep the correctness of the DIBE scheme, i.e., Invert in Lemma 4 (Item 4) can work ($\|\mathbf{T}_{\mathbf{F}_{id}}\| < p/(2\sqrt{m})$), where $\|\mathbf{T}_{\mathbf{F}_{id}}\| \leq \mathcal{O}(\sigma \cdot m)$ given by both $\mathsf{SampleBasisLeft}$ and $\mathsf{SampleBasisRight}$.
- ReRand (Lemma 5) in the security proof can operate ($\theta > \omega(\sqrt{\log \overline{m}})$, and $\theta' q/(2\theta q) > s_1([\mathbf{I}_m | \mathbf{R}'_{id^*}]^\top)$, where $s_1([\mathbf{I}_m | \mathbf{R}'_{id^*}]^\top) \leq (\beta + 1)$.
- Lemma 3 holds (q is super-polynomial and $\alpha/\theta = \mathsf{negl}(\lambda)$).
- $\Pr[\mathsf{Bad}_7] \leq 2m(2B+1)p/q = \mathsf{negl}(n) = n^{-\omega(1)}$, where $B = \theta' q\sqrt{n}$.

To satisfy the above requirements, we set the parameters in Figure 2. The private key size, ciphertext size and ciphertext expansion factor in our scheme are $\mathcal{O}(n^{2+3\eta})$, $\mathcal{O}(n^{1+\eta}\log(m\beta))$ and $\mathcal{O}(n^{\eta}\log(m\beta)/\log t)$ respectively. To optimize the ciphertext expansion factor, we can choose $t = m\beta$, which makes the ciphertext expansion factor to be $\mathcal{O}(n^{\eta})$.

Fig. 2. Parameter Selection of Generic DIBE Construction

Parameters	Description	Setting
λ	security parameter	
n	PK-lattice row dimension	$n = \lambda$
ℓ	the length of identity	$\ell = n$
$\overline{m}$	PK-matrix column number	$\mathcal{O}(n \log q)$
m	matrix column number	$m = \overline{m}$
σ	$\mathsf{SampleBasisLeft}$,$\mathsf{SampleBasisRight}$ width	$\max\{\beta, \sqrt{m}\} \cdot \omega(\sqrt{\log n})$
p	rounding parameter	$\mathcal{O}(\sigma \cdot m^{\frac{3}{2}})$
t	message space	$m\beta$
q	modulus	the prime nearest to 2^{n^η}, $0 < \eta < 1$
θ	error width	$\omega(\sqrt{\log n})$
θ'	error width	$\beta \cdot \omega(\sqrt{\log n})$

3.2 Security of DIBE

Theorem 1. *If $\mathcal{H} = (\mathcal{H}.\mathsf{Gen}, \mathcal{H}.\mathsf{Eval})$ is a $(1, v, \beta, \gamma, \delta)$ LPHF with high min-entropy from $\{0,1\}^\ell$ to $\mathbb{Z}_q^{n \times m}$, where γ is negligible, $\delta > 0$ is noticeable and independent of the modulus q, and large enough $v = \mathsf{poly}(n)$. Then, the above DIBE scheme DIBE is* PRIV1-ID-INDr*-secure with respect to $2^{-k \log t}$-hard-to-invert auxiliary inputs, assuming* $\mathrm{DLWE}_{q,n,\overline{m},\theta,\mathcal{H}_{\mathrm{hard}}}$ *is hard.*

According to Lemma 3, it is easy for us to get the following corollary.

Corollary 1. *If $\mathcal{H} = (\mathcal{H}.\mathsf{Gen}, \mathcal{H}.\mathsf{Eval})$ is a $(1, v, \beta, \gamma, \delta)$-LPHF with high min-entropy from $\{0,1\}^\ell$ to $\mathbb{Z}_q^{n \times m}$, where γ is negligible, $\delta > 0$ is noticeable and independent of the modulus q, and large enough $v = \mathsf{poly}(n)$. Then, the above DIBE scheme DIBE is* PRIV1-ID-INDr*-secure with respect to $2^{-k \log t}$-hard-to-invert auxiliary inputs, assuming* $\mathrm{DLWE}_{q,z,\overline{m},\alpha}$ *is hard, where $z \stackrel{\triangle}{=} \frac{k \log(t) - \omega(\log(\lambda))}{\log(q)}$.*

Proof of Theorem 1. Let $\mathcal{A}$ be a PPT adversary that breaks the PRIV1-ID-INDr-security with auxiliary inputs of the DIBE scheme. Moreover, let $\epsilon = \epsilon(\lambda)$ and $Q = Q(\lambda) \leq v$ be its advantage and the upper bound of the number of DIBE.KGen$(mpk, msk, \cdot)$ queries, respectively. And let $I^* = \{id^*, \{id_j\}_{j\in[Q]}\}$ denotes the challenge ID along with the queried IDs. For any distribution $\widehat{\mathcal{M}}$ over $\mathbb{Z}_t^n$, let $\mathcal{H}_{\text{hard}} = \{h\}$ be a set of $z^{-k\log(t)}$-hard-to-invert functions with respect to $\widehat{\mathcal{M}}$.

In order to prove the security of this DIBE scheme, we define a sequence of games. In each game, the challenger selects a uniform bit $\mathsf{coin} \xleftarrow{\$} \{0,1\}$, while the adversary $\mathcal{A}$ finally returns a guess bit coin' to the challenger. The challenger sets $\widehat{\mathsf{coin}} = \mathsf{coin}'$ in the orginal game, these values might be different in the latter games. In the following, we define X_i as the event that $\widehat{\mathsf{coin}} = \mathsf{coin}$.

Game$_0$: This game is the original PRIV1-ID-INDr game with auxiliary inputs. By definition, we have

$$|\Pr[X_0] - \frac{1}{2}| = |\Pr[\widehat{\mathsf{coin}} = \mathsf{coin}] - \frac{1}{2}| = |\Pr[\mathsf{coin}' = \mathsf{coin}] - \frac{1}{2}| = \epsilon.$$

Game$_1$: This game is identical to **Game$_0$** except that the challenger changes the setup and challenge phase as below.

Setup. It first generates a pair $(\mathbf{A}, \mathbf{T_A}) \xleftarrow{\$} \mathsf{TrapGen}(1^n, 1^{\overline{m}}, q)$, where $\mathbf{A} \in \mathbb{Z}_q^{n\times\overline{m}}$ and $\mathbf{T_A} \in \mathbb{Z}_q^{\overline{m}\times\overline{m}}$. Then, it computes $(K', td) \leftarrow \mathcal{H}.\mathsf{TrapGen}(\mathbf{A}, \mathbf{G})$. Finally, it outputs $mpk = (\mathbf{A}, K')$ and keeps the trapdoor td private.

Challenge Phase. The challenger directly uses (K', td) to generate the challenge ciphertext.

According to the property of statistically close trapdoor keys, we have $|\Pr[X_1] - \Pr[X_0]| < \mathsf{negl}(\lambda)$.

Game$_2$: This game is identical to **Game$_1$** except that the challenger performs the following additional step at the end of the game. The challenger first defines

$$\tau(\widehat{td}, \widehat{K}, I^*) = \begin{cases} 0 & \text{if } \widehat{\mathbf{S}}_{id^*} = 0 \text{ and } \widehat{\mathbf{S}}_{id_j} \text{ is invertible for all } j \in [Q] \\ 1 & \text{otherwise,} \end{cases}$$

where $(\widehat{\mathbf{R}}_{id^*}, \widehat{\mathbf{S}}_{id^*}) = \mathcal{H}.\mathsf{TrapEval}(\widehat{td}, \widehat{K}, id^*)$ and $(\widehat{\mathbf{R}}_{id_j}, \widehat{\mathbf{S}}_{id_j}) = \mathcal{H}.\mathsf{TrapEval}(\widehat{td}, \widehat{K}, id_j)$. Then, the challenger proceeds the following steps:

Abort Check : For (K', td) generated in the setup phase, if $\tau(td, K', I^*) = 1$, the challenger aborts the game and sets $\widehat{\mathsf{coin}} \xleftarrow{\$} \{0,1\}$ ignoring the output of $\mathcal{A}$. Otherwise, the following equation holds:

$$\mathcal{H}.\mathsf{TrapEval}(td, K', id) = \begin{cases} \mathbf{A}\mathbf{R}'_{id^*} & \text{if } id = id^* \\ \mathbf{A}\mathbf{R}'_{id} + \mathbf{S}'_{id}\mathbf{G} & \text{otherwise.} \end{cases}$$

Artificial Abort : Fix I^*, let p be the probability $p = \Pr[\tau(\widehat{td}, \widehat{K}, I^*) = 0]$ over the random choice of $(\widehat{td}, \widehat{K})$. The challenger samples $\mathcal{O}(\epsilon^{-2}\log(\epsilon^{-1})\delta^{-1}\log(\delta^{-1}))$

times the probability p by independently running $(\widehat{K}, \widehat{td}) \leftarrow \mathcal{H}.\mathsf{TrapGen}(\mathbf{A}, \mathbf{G})$ and evaluating $\tau(\widehat{td}, \widehat{K}, I^*)$ to compute an estimate p'. Then if $p' \geq \delta$, the challenger will abort with probability $\frac{p'-\delta}{p'}$ and sets $\widehat{\mathsf{coin}} \xleftarrow{\$} \{0,1\}$ ignoring the output of $\mathcal{A}$. Otherwise, when receiving coin' from $\mathcal{A}$, the challenger sets $\widehat{\mathsf{coin}} = \mathsf{coin}'$.

For $i \in \{2,3,4,5,6,7\}$, let $\widetilde{p}_i$ be the probability that the challenger does not abort in the abort check stage in $\mathbf{Game_i}$, and let p_i be the probability in the artificial abort stage of $\mathbf{Game_i}$ defined by $p_i = \Pr[\tau(\widehat{td}, \widehat{K}, I^*) = 0]$. Since the adversary might obtain some information of td from the challenge ciphertext, the probability $\widetilde{p}_i$ might not be equal to the probability p_i. Formally, let Γ_i be the absolute difference between $\widetilde{p}_i$ and p_i (i.e., $\Gamma_i = |\widetilde{p}_i - p_i|$). As we show in Lemma 2, we have

$$\left| \Pr[X_2] - \frac{1}{2} \right| \geq \frac{1}{2}\epsilon(\delta - \Gamma_2). \tag{1}$$

So as not to interrupt the proof of Theorem 1, we intentionally skip the proof for the time being.

$\mathbf{Game_3}$: This game is identical to **$\mathbf{Game_2}$** except that the challenger changes setup, phase 1 and 2, and challenge phase as below.

Setup. It first selects a random matrix $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times \overline{m}}$. Then, it computes $(K', td) \leftarrow \mathcal{H}.\mathsf{TrapGen}(\mathbf{A}, \mathbf{G})$. Finally, it outputs $mpk = (\mathbf{A}, K')$ and keeps the trapdoor td private.

Phase 1. When receiving the private key query with identity id, the challenger first computes $(\mathbf{R}'_{id}, \mathbf{S}'_{id}) = \mathcal{H}.\mathsf{TrapEval}(td, K', id)$. If $\mathbf{S}'_{id}$ is not invertible, the challenger aborts the game and sets $\widehat{\mathsf{coin}} \xleftarrow{\$} \{0,1\}$. Otherwise, it computes $sk_{id} = \mathsf{SampleBasisRight}(\mathbf{A}, \mathbf{G}, \mathbf{R}'_{id}, \mathbf{S}'_{id}, \mathbf{T}_{\mathbf{G}}, \sigma)$ and sends sk_{id} to $\mathcal{A}$.

Challenge Phase. The challenger computes $(\mathbf{R}'_{id^*}, \mathbf{S}'_{id^*}) = \mathcal{H}.\mathsf{TrapEval}(td, K', id^*)$. If $\mathbf{S}_{id^*} \neq \mathbf{0}$, the challenger aborts the game and sets $\widehat{\mathsf{coin}} \xleftarrow{\$} \{0,1\}$. Otherwise, for $\mathbf{m} \xleftarrow{\$} \widehat{\mathcal{M}}$, the challenger computes

$$\widehat{\mathbf{c}} = \mathbf{F}_{id^*}^{\top}\mathbf{m} = \begin{bmatrix} \mathbf{A}^{\top}\mathbf{m} \\ (\mathbf{R}'_{id^*})^{\top}\mathbf{A}^{\top}\mathbf{m} \end{bmatrix} \in \mathbb{Z}_q^{m'}.$$

Then, the challenger sets $\mathbf{c}_0^* = \lfloor \widehat{\mathbf{c}} \rceil_p$. Finally, the challenger returns $(\mathbf{c}^*_{\mathsf{coin}}, h(\mathbf{m}))$ to the adversary $\mathcal{A}$.

Phase 2. The challenger responds as in Phase 1, when receiving the private key query with identity $id \neq id^*$.

It is easy to see that

$$\Pr[X_3] = \Pr[X_2] \quad \text{and} \quad \Pr[\Gamma_3] = \Pr[\Gamma_2]. \tag{2}$$

$\mathbf{Game_4}$: In this game, the challenger changes the way that the challenge ciphertext is created when $\mathsf{coin} = 0$.

Challenge Phase. The challenger computes $(\mathbf{R}'_{id^*}, \mathbf{S}'_{id^*}) = \mathcal{H}.\mathsf{TrapEval}(td, K', id^*)$. If $\mathbf{S}_{id^*} \neq \mathbf{0}$, the challenger aborts the game and sets $\widehat{\mathsf{coin}} \xleftarrow{\$} \{0,1\}$. Otherwise, for $\mathbf{m} \xleftarrow{\$} \widehat{\mathcal{M}}$, the challenger chooses $\mathbf{e_1} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^{\overline{m}}, \theta' q}, \mathbf{e_2} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^{m}, \theta' q}$ and

computes

$$\widehat{\mathbf{c}} = \widehat{\mathbf{c}}_{\mathbf{1}} + \widehat{\mathbf{c}}_{\mathbf{2}} = \begin{bmatrix} \mathbf{A}^\top \mathbf{m} \\ (\mathbf{R}'_{id^*})^\top \mathbf{A}^\top \mathbf{m} \end{bmatrix} + \begin{bmatrix} \mathbf{e_1} \\ \mathbf{e_2} \end{bmatrix} = \begin{bmatrix} \mathbf{A}^\top \mathbf{m} + \mathbf{e_1} \\ (\mathbf{R}'_{id^*})^\top \mathbf{A}^\top \mathbf{m} + \mathbf{e_2} \end{bmatrix}.$$

Then, the challenger computes $\mathbf{c}_0^* = \lfloor \widehat{\mathbf{c}} \rceil_p$. Finally, the challenger returns $(\mathbf{c}^*_{\mathsf{coin}}, h(\mathbf{m}))$ to the adversary $\mathcal{A}$.

Before analyzing the difference between Game_3 and Game_4, we first define a "bad event" as follows: $\mathsf{Bad}_4 \stackrel{\triangle}{=} \lfloor \widehat{\mathbf{c}}_1 + [-B, B]^{m'} \rceil_p \neq \lfloor \widehat{\mathbf{c}}_1 \rceil_p$, where $B = \theta' q \sqrt{n}$. For $i \in \{5, 6, 7\}$, similar event Bad_i can also be defined in Game_i. If Bad_4 does not occur for some $\widehat{\mathbf{c}}_1$, then we have

$$\lfloor \widehat{\mathbf{c}} \rceil_p = \begin{bmatrix} \lfloor \mathbf{A}^\top \mathbf{m} + \mathbf{e_1} \rceil_p \\ \lfloor (\mathbf{R}'_{id^*})^\top \mathbf{A}^\top \mathbf{m} + \mathbf{e_2} \rceil_p \end{bmatrix} = \begin{bmatrix} \lfloor \mathbf{A}^\top \mathbf{m} \rceil_p \\ \lfloor (\mathbf{R}'_{id^*})^\top \mathbf{A}^\top \mathbf{m} \rceil_p \end{bmatrix} = \lfloor \mathbf{F}^\top_{id^*} \mathbf{m} \rceil_p.$$

It immediately follows that for any adversary $\mathcal{A}$

$$|\Pr[X_4] - \Pr[X_3]| \leq \Pr[\mathsf{Bad}_4] \quad \text{and} \quad |\Pr[\Gamma_4] - \Pr[\Gamma_3]| \leq \Pr[\mathsf{Bad}_4]. \tag{3}$$

$\mathbf{Game_5}$: In this game, the challenger changes the way that the challenge ciphertext is created when $\mathsf{coin} = 0$.

Challenge Phase. The challenger computes $(\mathbf{R}'_{id^*}, \mathbf{S}'_{id^*}) = \mathcal{H}.\mathsf{TrapEval}(td, K', id^*)$.

If $\mathbf{S}'_{id^*} \neq \mathbf{0}$, the challenger aborts the game and sets $\widehat{\mathsf{coin}} \stackrel{\$}{\leftarrow} \{0, 1\}$. Otherwise, when $\mathsf{coin} = 0$, the challenger first picks $\mathbf{m} \stackrel{\$}{\leftarrow} \widehat{\mathcal{M}}$ and $\mathbf{e} \stackrel{\$}{\leftarrow} \mathcal{D}_{\mathbb{Z}^{\overline{m}}, \theta q}$, and computes $\mathbf{b} = \mathbf{A}^\top \mathbf{m} + \mathbf{e}$. It runs the algorithm ReRand to get $\widehat{\mathbf{c}}$, i.e.,

$$\widehat{\mathbf{c}} = \mathsf{ReRand}\left(\begin{bmatrix} \mathbf{I}_{\overline{m}} \\ (\mathbf{R}'_{id^*})^\top \end{bmatrix}, \mathbf{b}, \theta q, \frac{\theta' q}{2\theta q} \right)$$

, where $\mathbf{I}_{\overline{m}}$ is the unit matrix of size $\overline{m} \times \overline{m}$. Then, the challenger computes $\mathbf{c}_0^* = \lfloor \widehat{\mathbf{c}} \rceil_p$. Finally, the challenger returns $(\mathbf{c}^*_{\mathsf{coin}}, h(\mathbf{m}))$ to the adversary $\mathcal{A}$.

According to the property of the algorithm ReRand in the Lemma 5, we have

$$\Pr[X_5] = \Pr[X_4] \quad \text{and} \quad \Pr[\Gamma_5] = \Pr[\Gamma_4] \quad \text{and} \quad \Pr[\mathsf{Bad}_5] = \Pr[\mathsf{Bad}_4]. \tag{4}$$

$\mathbf{Game_6}$: In this game, the challenger changes the way that the challenge ciphertext is created when $\mathsf{coin} = 0$.

Challenge Phase. The challenger computes $(\mathbf{R}'_{id^*}, \mathbf{S}'_{id^*}) = \mathcal{H}.\mathsf{TrapEval}(td, K', id^*)$.

If $\mathbf{S}'_{id^*} \neq \mathbf{0}$, the challenger aborts the game and sets $\widehat{\mathsf{coin}} \stackrel{\$}{\leftarrow} \{0, 1\}$. Otherwise, when $\mathsf{coin} = 0$, the challenger first picks $\mathbf{v} \stackrel{\$}{\leftarrow} \mathbb{Z}_q^{\overline{m}}, \mathbf{e} \stackrel{\$}{\leftarrow} \mathcal{D}_{\mathbb{Z}^{\overline{m}}, \theta q}$ and sets $\mathbf{b} = \mathbf{v} + \mathbf{e}$. Then, it computes

$$\widehat{\mathbf{c}} = \mathsf{ReRand}\left(\begin{bmatrix} \mathbf{I}_{\overline{m}} \\ (\mathbf{R}'_{id^*})^\top \end{bmatrix}, \mathbf{b}, \theta q, \frac{\theta' q}{2\theta q} \right) \in \mathbb{Z}_q^{m'}.$$

Then, the challenger computes $\mathbf{c}_0^* = \lfloor \widehat{\mathbf{c}} \rceil_p$. Finally, the challenger returns $(\mathbf{c}^*_{\mathsf{coin}}, h(\mathbf{m}))$ to the adversary $\mathcal{A}$.

We construct an algorithm $\mathcal{B}$ against the problem $\mathrm{DLWE}_{q,n,\overline{m},\theta,\mathcal{H}_{\mathrm{hard}}}$ as follows. Given the problem instance of LWE $(\mathbf{A}, \mathbf{b} = \mathbf{v} + \mathbf{e}) \in \mathbb{Z}_q^{n \times \overline{m}} \times \mathbb{Z}_q^{\overline{m}}$, where $\mathbf{e} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^{\overline{m}},\theta q}$. The task of $\mathcal{B}$ is to distinguish whether $\mathbf{v} = \mathbf{A}^\top \mathbf{m}$ for $\mathbf{m} \xleftarrow{\$} \mathbb{Z}_t^n$ or $\mathbf{v} \xleftarrow{\$} \mathbb{Z}_q^{\overline{m}}$. This subtle change from the standard $\mathrm{DLWE}_{q,n,\overline{m},\theta,\mathcal{H}}$ is done only for convenience of the proof. $\mathcal{B}$ can simulate the security game for the adversary $\mathcal{A}$. If $\mathbf{v} = \mathbf{A}^\top \mathbf{m}$, the view of $\mathcal{A}$ corresponds to $\mathbf{Game}_5$; otherwise, for $\mathbf{v} \xleftarrow{\$} \mathbb{Z}_q^{\overline{m}}$, the view of $\mathcal{A}$ corresponds to $\mathbf{Game}_6$. As a result, we get that

$$\begin{aligned} &|\Pr[X_6] - \Pr[X_5]| \leq \mathrm{DLWE}_{q,n,\overline{m},\theta,\mathcal{H}_{\mathrm{hard}}}, \\ &|\Pr[\Gamma_6] - \Pr[\Gamma_5]| \leq \mathrm{DLWE}_{q,n,\overline{m},\theta,\mathcal{H}_{\mathrm{hard}}}, \\ &|\Pr[\mathsf{Bad}_6] - \Pr[\mathsf{Bad}_5]| \leq \mathrm{DLWE}_{q,n,\overline{m},\theta,\mathcal{H}_{\mathrm{hard}}}. \end{aligned} \tag{5}$$

$\mathbf{Game_7}$: In this game, the challenger changes the way that the challenge ciphertext is created when $\mathsf{coin} = 0$.

Challenge Phase. The challenger computes $(\mathbf{R}'_{id^*}, \mathbf{S}'_{id^*}) = \mathcal{H}.\mathsf{TrapEval}(td, K', id^*)$.

If $\mathbf{S}'_{id^*} \neq \mathbf{0}$, the challenger aborts the game and sets $\widehat{\mathsf{coin}} \xleftarrow{\$} \{0,1\}$. Otherwise, when $\mathsf{coin} = 0$, the challenger computes

$$\widehat{\mathbf{c}} = \widehat{\mathbf{c}}_{\mathbf{1}} + \widehat{\mathbf{c}}_{\mathbf{2}} = \begin{bmatrix} \mathbf{v} \\ (\mathbf{R}'_{id^*})^\top \mathbf{v} \end{bmatrix} + \begin{bmatrix} \mathbf{e_1} \\ \mathbf{e_2} \end{bmatrix} = \begin{bmatrix} \mathbf{v} + \mathbf{e_1} \\ (\mathbf{R}'_{id^*})^\top \mathbf{v} + \mathbf{e_2} \end{bmatrix},$$

instead of running the algorithm ReRand, where $\mathbf{e}_1 \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^{\overline{m}},\theta' q}$, $\mathbf{e}_2 \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^m,\theta' q}$, $\mathbf{v} \xleftarrow{\$} \mathbb{Z}_q^{\overline{m}}$. Then, the challenger computes $\mathbf{c}_0^* = \lfloor \widehat{\mathbf{c}} \rceil_p$. Finally, the challenger returns $(\mathbf{c}^*_{\mathsf{coin}}, h(\mathbf{m}))$ to the adversary $\mathcal{A}$.

According to the property of the algorithm ReRand, we have

$$\Pr[X_7] = \Pr[X_6] \quad \text{and} \quad \Pr[\Gamma_7] = \Pr[\Gamma_6] \quad \text{and} \quad \Pr[\mathsf{Bad}_7] = \Pr[\mathsf{Bad}_6]. \tag{6}$$

Because for $id^* \in \mathrm{ID}$ the statistical distance between $(\mathbf{A}, K', \mathbf{v}, (\mathbf{R}'_{id^*})^\top \mathbf{v})$ and $(\mathbf{A}, K', \mathbf{v}, \mathbf{u})$ is negligible in λ, where $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$, $\widehat{\mathbf{c}}_{\mathbf{1}}$ is statistically close to uniform distribution over $\mathbb{Z}_q^{m'}$, therefore for uniform $\widehat{\mathbf{c}}_{\mathbf{1}}$,

$$\Pr[\mathsf{Bad}_7] \leq 2m(2B+1)p/q = \mathsf{negl}(\lambda), \tag{7}$$

by assumption on q and θ'. In the meantime, because $\widehat{\mathbf{c}}$ is statistically close to uniform distribution over $\mathbb{Z}_q^{m'}$, we can get that

$$\Pr[X_7] = \frac{1}{2} \quad \text{and} \quad \Pr[\Gamma_7] = 0. \tag{8}$$

Summing up equations (1)-(8), we can get

$$\mathrm{DLWE}_{q,n,\overline{m},\theta,\mathcal{H}_{\mathrm{hard}}} \geq \frac{\epsilon\delta}{6} - \mathsf{negl}(\lambda).$$

□

In order to prove Theorem 1, we should prove that equation (1) holds. We will use Lemma 28 in the full version of the work [1], which is described as follows.

Lemma 1 ([1]). *Let I^* be a $(Q+1)$-ID tuple $\{id^*, \{id_j\}_{j\in[Q]}\}$ denoted the challenge ID along with the queried ID's, and $\eta(I^*)$ define the probability that an abort does not happen in* $\mathbf{Game_2}$. *Let $\eta_{max} = \max\ \eta(I^*)$ and $\eta_{min} = \min\ \eta(I^*)$. For $i = 1, 2$, we set X_i be the event that $\widehat{\mathsf{coin}} = \mathsf{coin}$ at the end of* $\mathbf{Game_i}$. *Then,*

$$\left|\Pr[X_2] - \frac{1}{2}\right| \geq \eta_{min}\left|\Pr[X_1] - \frac{1}{2}\right| - \frac{1}{2}(\eta_{max} - \eta_{min}).$$

Lemma 2. *If $\mathcal{H}$ is a $(1, v, \beta, \gamma, \delta)$-LPHF with high min-entropy and $Q \leq v$, then $\left|\Pr[X_2] - \frac{1}{2}\right| \geq \frac{1}{2}\epsilon(\delta - \Gamma_2)$.*

Proof. According to Lemma 1, we only need to compute η_{max}, η_{min} and $\eta_{max} - \eta_{min}$. By the definition of $\widetilde{p}_2$ and p_2 in $\mathbf{Game_2}$, we have $\eta(I^*) = \widetilde{p}_2\frac{\delta}{p'}$, where p' is an estimate of p_2. Since the challenger always samples $\mathcal{O}(\epsilon^{-2}\log(\epsilon^{-1})\delta^{-1}\log(\delta^{-1}))$ times the probability p_2 to compute p', according to the Chernoff bounds, we have $\Pr[p' > p_2(1+\frac{\epsilon}{8})] < \delta\frac{\epsilon}{8}$ and $\Pr[p' < p_2(1-\frac{\epsilon}{8})] < \delta\frac{\epsilon}{8}$. As a result, the following equations hold

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

Finally, we have $\left|\Pr[X_2] - \frac{1}{2}\right| \geq \frac{7\delta\widetilde{p}_2}{9p_2}\cdot\epsilon - \frac{1}{2}\cdot\frac{16\epsilon\delta\widetilde{p}_2}{63p_2} \geq \frac{\epsilon\delta(p_2-\Gamma_2)}{2p_2} \geq \frac{1}{2}\epsilon(\delta-\Gamma_2)$.□□

4 Constructions of LPHFs with High Min-Entropy

In [22], Zhang et al. proved that the IBE schemes in [1, 15, 22] implies instantiations of LPHFs with high min-entropy. In fact, the IBE scheme in [4] also imply an instantiation of LPHF with high min-entropy.

In this section, we show that LPHFs with high min-entropy can be constructed from partitioning functions with compatible algorithms [20]. Moreover, we prove that the adaptively secure and anonymous IBE schemes in [19, 14] naturally imply instantiations of LPHFs with high min-entropy. In a word, the adaptively secure and anonymous IBE schemes in [1, 15, 22, 4, 19, 14, 20] naturally imply instantiations of LPHFs with high min-entropy.

4.1 From Partitioning Functions with Compatible Algorithms [20]

Let $\mathcal{F}_{\mathrm{PF}} : \mathcal{K}_{\mathrm{PF}} \times \mathrm{ID} \to \{0,1\}$ be a partitioning function with associating δ_{PF}-compatible algorithms $(\mathsf{Encode}, \mathsf{PubEval}, \mathsf{TrapEval})$ (See Appendix A). We assume $\mathrm{ID} = \{0,1\}^\ell$. Now, we show how to construct a $(1, v, \beta, \gamma, \delta)$-LPHF with high min-entropy from the partitioning function $\mathcal{F}_{\mathrm{PF}} : \mathcal{K}_{\mathrm{PF}} \times \mathrm{ID} \to \{0,1\}$.

A hash function $\mathcal{H} : \mathrm{ID} \to \mathbb{Z}_q^{n\times m}$ consists of two algorithms $(\mathcal{H}.\mathsf{Gen}, \mathcal{H}.\mathsf{Eval})$ which are defined as follows:

- $\mathcal{H}.\mathsf{Gen}(1^\lambda) \to K_{\mathrm{lphf}}$: It randomly chooses matrices $\mathbf{B}_1, \cdots, \mathbf{B}_u \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, $\mathbf{B}_0 \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, and returns these $u+1$ matrices, i.e., $K_{\mathrm{lphf}} := \{\mathbf{B}_1, \cdots, \mathbf{B}_u, \mathbf{B}_0\}$.
- $\mathcal{H}.\mathsf{Eval}(K_{\mathrm{lphf}}, id \in \mathrm{ID}) \to \mathbf{Z} \in \mathbb{Z}_q^{n\times m}$: For $id \in \mathrm{ID}$, it first gets $\mathbf{B}_{id}$ by running the algorithm $\mathsf{PubEval}(id \in \mathrm{ID}, \{\mathbf{B}_i \in \mathbb{Z}_q^{n\times m}\}_{i\in[u]})$. Then, it returns $\mathbf{Z} = \mathbf{B}_0 + \mathbf{B}_{id}$.

The associating algorithms $\mathcal{H}.\mathsf{TrapGen}$ and $\mathcal{H}.\mathsf{TrapEval}$ are defined as follows.

- $\mathcal{H}.\mathsf{TrapGen}(1^\lambda, \mathbf{A}, \mathbf{G}) \to (K'_{\mathrm{lphf}}, td)$: It first computes $K_{\mathrm{PF}} \xleftarrow{\$} \mathsf{PrtSmp}(1^\lambda, Q, \epsilon)$. Then, it gets $k \in \{0,1\}^u$ by operating the algorithm $\mathsf{Encode}(K_{\mathrm{PF}})$. Finally, it randomly chooses matrices $\mathbf{R}_1, \cdots, \mathbf{R}_u, \mathbf{R}_0 \xleftarrow{\$} \{-1,1\}^{m\times m}$, and returns $K'_{\mathrm{lphf}} := \{\mathbf{A}\mathbf{R}_1 + k_1\mathbf{G}, \cdots, \mathbf{A}\mathbf{R}_u + k_u\mathbf{G}, \mathbf{A}\mathbf{R}_0\}$ and $td = \{K_{\mathrm{PF}}, \mathbf{R}_1, \cdots, \mathbf{R}_u, \mathbf{R}_0\}$.
- $\mathcal{H}.\mathsf{TrapEval}(td, K'_{\mathrm{lphf}}, id \in \mathrm{ID}) \to (\mathbf{R}'_{id} \in \mathbb{Z}_q^{m\times m}, \mathbf{S}'_{id} \in \mathbb{Z}_q^{n\times n})$: For $id \in \mathrm{ID}$, it defines $\mathbf{R}'_{id} = \mathbf{R}_0 + \mathsf{TrapEval}(K_{\mathrm{PF}}, id, \mathbf{A}, \{\mathbf{R}_i\}_{i\in[u]})$ and $\mathbf{S}'_{id} = \mathcal{F}_{\mathrm{PF}}(K_{\mathrm{PF}}, id) \cdot \mathbf{I}_n$, where $\mathbf{I}_n$ denotes the identity matrix of $n \times n$. In this case, $s_1(\mathbf{R}'_{id}) \le \sqrt{m} \cdot \sqrt{2m} \cdot \|\mathbf{R} + \mathsf{TrapEval}(K_{\mathrm{PF}}, id, \mathbf{A}, \{\mathbf{R}_i\}_{i\in[u]})\|_\infty \le \sqrt{2}m \cdot (1 + \delta_{\mathrm{PF}})$.

Now, we show that this construction satisfies the following properties:

- Correctness: $\mathcal{H}.\mathsf{Eval}(K'_{\mathrm{lphf}}, id) = \mathbf{B}_0 + \mathsf{PubEval}(id, \{\mathbf{A}\mathbf{R}_i + k_i\mathbf{G}\}_{i\in[u]}) = \mathbf{A}\mathbf{R}_0 + (\mathbf{A} \cdot \mathsf{TrapEval}(K, id, \mathbf{A}, \{\mathbf{R}_i\}_{i\in[u]}) + \mathcal{F}_{\mathrm{PF}}(K_{\mathrm{PF}}, id) \cdot \mathbf{G}) = \mathbf{A}\mathbf{R}'_{id} + \mathbf{S}'_{id}\mathbf{G}$.
- Statistically close trapdoor keys: According to the Leftover Hash Lemma, the statistical distance between the distributions $\{\mathbf{A}, \mathbf{B}_1, \cdots, \mathbf{B}_u, \mathbf{B}_0\}$ and $\{\mathbf{A}, \mathbf{A}\mathbf{R}_1 + k_1\mathbf{G}, \cdots, \mathbf{A}\mathbf{R}_u + k_u\mathbf{G}, \mathbf{A}\mathbf{R}_0\}$ is negligible. As a result, the statistical distance between $\{\mathbf{A}, K_{\mathrm{lphf}}\}$ and $\{\mathbf{A}, K'_{\mathrm{lphf}}\}$ is negligible, i.e., $\gamma = \mathsf{negl}(\lambda)$.
- Well-distributed hidden matrices: For all $(K'_{\mathrm{LPHF}}, td) \leftarrow \mathcal{H}.\mathsf{TrapGen}(1^\lambda, \mathbf{A}, \mathbf{G})$, any inputs $id^*, id_1, \cdots, id_v$ such that $id^* \neq id_j$ for any $j \in [v]$. Then,

$$\begin{aligned}
&\Pr[\mathbf{S}'_{id^*} = \mathbf{0} \wedge \mathbf{S}'_{id_1}, \cdots, \mathbf{S}'_{id_v} \in \mathcal{I}_n] \\
=&\Pr[\mathcal{F}_{\mathrm{PF}}(K_{\mathrm{PF}}, id^*) = 0 \wedge \mathcal{F}_{\mathrm{PF}}(K_{\mathrm{PF}}, id_1) = \cdots = \mathcal{F}_{\mathrm{PF}}(K_{\mathrm{PF}}, id_v) = 1] \\
\ge&\gamma_{\min}(\lambda).
\end{aligned}$$

In a word, this construction is a $(1, v, \sqrt{2}m \cdot (1 + \delta_{\mathrm{PF}}), \mathsf{negl}(\lambda), \gamma_{\min}(\lambda))$-LPHF. Finally, we show that this LPHF possesses the property 1, i.e., with high min-entropy.

- For any $(K'_{\mathrm{lphf}}, td) \leftarrow \mathcal{H}.\mathsf{TrapGen}(1^\lambda, \mathbf{A}, \mathbf{G})$, any $id \in \mathrm{ID}$ and its corresponding $(\mathbf{R}'_{id}, \mathbf{S}'_{id}) = \mathcal{H}.\mathsf{TrapEval}(td, K'_{\mathrm{lphf}}, id)$, the following distributions are statistically close:

$$\begin{aligned}
&(\mathbf{A}, K'_{\mathrm{lphf}}, \mathbf{v}, (\mathbf{R}'_{id})^\top \mathbf{v}) \\
=&(\mathbf{A}, \mathbf{A}\mathbf{R}_0, \{\mathbf{A}\mathbf{R}_i + k_i\mathbf{G}\}, \mathbf{v}, (\mathbf{R} + \mathsf{TrapEval}(K, id, \mathbf{A}, \{\mathbf{R}_i\}_{i\in[u]}))^\top \mathbf{v}) \\
\approx&(\mathbf{A}, \mathbf{B}_0, \{\mathbf{A}\mathbf{R}_i + k_i\mathbf{G}\}, \mathbf{v}, \mathbf{u}) \\
\approx&(\mathbf{A}, K'_{\mathrm{lphf}}, \mathbf{v}, \mathbf{u}),
\end{aligned}$$

where $\mathbf{B}_0 \xleftarrow{\$} \mathbb{Z}_q^{n\times m}, \mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m, \mathbf{v} \xleftarrow{\$} \mathbb{Z}_q^m$. It can been seen that the second and the third distributions are $\mathsf{negl}(\lambda)$-close, by applying Leftover Hash Lemma for $[\mathbf{A}|\mathbf{v}^\top] \in \mathbb{Z}_q^{(n+1)\times m}$ and $\mathbf{R}$. □

4.2 From Yam16 [19] and KY16 [14]

In [19], Yamada proposed an adaptively secure and anonymous IBE with asymptotically short parameters. In particular, the master public key consists of $\mathcal{O}(\ell^{1/2})$ basic matrices. In this part, we show that their construction implies a LPHF with high min-entropy. For simplicity, we denotes it by $\mathcal{H}_{\text{Yam16}} : \text{ID} \to \mathbf{Z} \in \mathbb{Z}_q^{n\times m}$, where $\text{ID} = \{0,1\}^\ell$. In their construction, there exists an efficiently computable injective map $\mathcal{S}$ that maps an element $id \in \text{ID}$ to a subset $\mathcal{S}(id)$ of $[1,t]^2$, where $t = \lceil\sqrt{\ell}\rceil$. The algorithms $(\mathcal{H}_{\text{Yam16}}.\mathsf{Gen}, \mathcal{H}_{\text{Yam16}}.\mathsf{Eval})$ are defined as below.

- $\mathcal{H}_{\text{Yam16}}.\mathsf{Gen}(1^\lambda) \to K$: It picks random matrices $\mathbf{B}_0 \xleftarrow{\$} \mathbb{Z}_q^{n\times m}, \mathbf{B}_{i,j} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ for $(i,j) \in [2]\times[t]$ and returns $K = (\mathbf{B}_0, \{\mathbf{B}_{i,j}\}_{(i,j)\in[2]\times[t]})$.
- $\mathcal{H}_{\text{Yam16}}.\mathsf{Eval}(K, id) \to \mathbf{Z} \in \mathbb{Z}_q^{n\times m}$: For all $id \in \text{ID}$, the algorithm $\mathcal{H}_{\text{Yam16}}.\mathsf{Eval}$ is defined as follows,

$$\mathcal{H}_{\text{Yam16}}.\mathsf{Eval}(K, id) = \mathbf{B}_0 + \sum_{(j_1,j_2)\in\mathcal{S}(id)} \mathbf{B}_{1,j_1} \cdot \mathbf{G}^{-1}(\mathbf{B}_{2,j_2}) \in \mathbb{Z}_q^{n\times m}.$$

The associating algorithms $\mathcal{H}_{\text{Yam16}}.\mathsf{TrapGen}$ and $\mathcal{H}_{\text{Yam16}}.\mathsf{TrapEval}$ are defined as

- $\mathcal{H}_{\text{Yam16}}.\mathsf{TrapGen}(1^\lambda, \mathbf{A}, \mathbf{G}) \to (K', td)$: It first selects random elements $y_0 \xleftarrow{\$} [-4(\ell+1)n^{2c}+1, 0]$ and $y_{i,j} \xleftarrow{\$} [1, 2n^c]$, where $c = c_1+c_2$ and c_1, c_2 satisfy that $\frac{n^{c_1}}{2} \geq Q+1$ and $n^{-c_2} \leq \epsilon$. Then, it randomly chooses matrices $\mathbf{R}_0, \mathbf{R}_{i,j} \xleftarrow{\$} \{-1,1\}^{m\times m}$. Finally, it computes $\mathbf{B}_0 = \mathbf{A}\mathbf{R}_0 + y_0\mathbf{G}$ and $\mathbf{B}_{i,j} = \mathbf{A}\mathbf{R}_{i,j} + y_{i,j}\mathbf{G}$ returns $K' = \{\mathbf{B}_0, \mathbf{B}_{i,j}\}$ and $td = \{y_0, y_{i,j}, \mathbf{R}_0, \mathbf{R}_{i,j}\}$.
- $\mathcal{H}_{\text{Yam16}}.\mathsf{TrapEval}(td, K', id) \to (\mathbf{R}'_{id} \in \mathbb{Z}_q^{m\times m}, \mathbf{S}'_{id} \in \mathbb{Z}_q^{n\times n})$: For $id \in \text{ID}$,

$$\mathbf{R}'_{id} = \mathbf{R}_0 + \sum_{(j_1,j_2)\in\mathcal{S}(id)} \left(\mathbf{R}_{1,j_1}\mathbf{G}^{-1}(\mathbf{B}_{2,j_2}) + y_{1,j_1}\mathbf{R}_{2,j_2}\right),$$
$$\mathbf{S}'_{id} = (y_0 + \sum_{(j_1,j_2)\in\mathcal{S}(id)} y_{1,j_1} \cdot y_{1,j_2}) \cdot \mathbf{I}_n.$$

  In this case, $s_1(\mathbf{R}'_{id}) \leq m(1+4\ell n^c)$.

Now, we show that this construction satisfies the following properties:

- Correctness: It is easy to verify that $\mathcal{H}.\mathsf{Eval}(K', id) = \mathbf{A}\mathbf{R}'_{id} + \mathbf{S}'_{id}\mathbf{G}$.
- Statistically close trapdoor keys: According to the Leftover Hash Lemma, the statistical distance between the distributions $(\mathbf{A}, \mathbf{B}_0, \{\mathbf{B}_{i,j}\}_{(i,j)\in[2]\times[t]})$ and $(\mathbf{A}, \mathbf{A}\mathbf{R}_0 + y_0\mathbf{G}, \{\mathbf{A}\mathbf{R}_{i,j} + y_{i,j}\mathbf{G}\}_{(i,j)\in[2]\times[t]})$ is negligible.

- Well-distributed hidden matrices: For all $(K', td) \leftarrow \mathcal{H}.\mathsf{TrapGen}(1^\lambda, \mathbf{A}, \mathbf{G})$, any inputs $id^*, id_1, \cdots, id_v$ such that $id^* \neq id_j$ for any $j \in [v]$. Then,

$$\begin{aligned}&\Pr[\mathbf{S}'_{id^*} = \mathbf{0} \wedge \mathbf{S}'_{id_1}, \cdots, \mathbf{S}'_{id_v} \in \mathcal{I}_n] \\ =&\Pr[\mathcal{F}_{\mathbf{y}}(id^*) = 0 \wedge \mathcal{F}_{\mathbf{y}}(id_1) \neq 0 \cdots = \mathcal{F}_{\mathbf{y}}(id_v) \neq 0] \geq \frac{\epsilon^2}{32(\ell+1)Q^2},\end{aligned}$$

  where $\mathcal{F}_{\mathbf{y}}(id) = y_0 + \sum_{(j_1,j_2)\in\mathcal{S}(id)} y_{1,j_1} \cdot y_{1,j_2}$.

In a word, this construction is a $(1, v, m(1 + 4\ell n^c), \mathsf{negl}(\lambda), \frac{\epsilon^2}{32(\ell+1)Q^2})$-LPHF. Then, we show that $\mathcal{H}_{\mathrm{Yam16}}$ is a LPHF which possess the properties 1, i.e., with high min-entropy.

- **Property 1.** For any $(K', td) \leftarrow \mathcal{H}_{\mathrm{Yam16}}.\mathsf{TrapGen}(1^\lambda, \mathbf{A}, \mathbf{G})$, any $id \in \mathrm{ID}$ and its corresponding $(\mathbf{R}'_{id}, \mathbf{S}'_{id}) = \mathcal{H}_{\mathrm{Yam16}}.\mathsf{TrapEval}(td, K', id)$, the following distributions are statistically close:

$$\begin{aligned}(\mathbf{A}, K', \mathbf{v}, (\mathbf{R}'_{id})^\top \mathbf{v}) &= (\mathbf{A}, \mathbf{A}\mathbf{R}_0 + y_0\mathbf{G}, \{\mathbf{A}\mathbf{R}_{i,j} + y_{i,j}\mathbf{G}\}, \mathbf{v}, (\mathbf{R}_0 + \mathbf{R}')^\top \mathbf{v}) \\ &\approx (\mathbf{A}, \mathbf{B}_0, \{\mathbf{A}\mathbf{R}_{i,j} + y_{i,j}\mathbf{G}\}, \mathbf{v}, \mathbf{u}) \\ &= (\mathbf{A}, K', \mathbf{v}, \mathbf{u}),\end{aligned}$$

  where $\mathbf{R}' = \sum_{(j_1,j_2)\in\mathcal{S}(id)} \left(\mathbf{R}_{1,j_1}\mathbf{G}^{-1}(\mathbf{B}_{2,j_2}) + y_{1,j_1}\mathbf{R}_{2,j_2}\right)$, $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m, \mathbf{v} \xleftarrow{\$} \mathbb{Z}_q^m$. The second and the third distributions are $\mathsf{negl}(\lambda)$-close, by applying the Leftover Hash Lemma for $[\mathbf{A}^\top|\mathbf{v}] \in \mathbb{Z}_q^{(n+1)\times m}$ and $\mathbf{R}_0$. □

Remark 2. The subsequent work by Katsumata and Yamada [14] showed that for the ring version of Yamada's scheme [19], it is possible to reduce the magnitude of $s_1(\mathbf{R}'_{id})$ (which inluences the selection of modulus q). We do not see any obstacle preventing us from constructing a programmable hash function with high min entropy from ideal lattices, according to the IBE scheme of [14].

5 Instantiations of Generic DIBE construction

As mentioned in section 4, there are many LPHFs with high min-entropy in [1, 15, 22, 4, 20, 19, 14]. However, except the LPHF with high min-entropy [1] used by Xie et al. [18] to construct DIBE scheme, there only exist another four $(1, v, \beta, \gamma, \delta)$ LPHFs with high min-entropy which satisfy the requirement that δ is independent of the modulus q, under the LWE assumption. These four LPHFs with high min-entropy are briefly described in the following.

Zhang et al. [22] constructed one LPHF with high min-entropy, and using this LPHF with high min-entropy they presented an adaptively secure IBE scheme with more compact public parameters.

(1) $\mathcal{H}_{\mathrm{ZCZ16}}$ in ZCZ16 [22]: a $(1, v, \mu v\ell\overline{m}^{1.5} \cdot \omega(\sqrt{\log \overline{m}}), \mathsf{negl}(\lambda), \frac{1}{N})$ LPHF with high min-entropy, where $N \leq 16v^2\ell$ and $\mu = \lceil \log N \rceil$. Additionally, the key of $\mathcal{H}_{\mathrm{ZCZ16}}$ only consists of $\mu = \lceil \log N \rceil$ matrices.

In [20], Yamada elaborately constructed two partitioning functions with compatible algorithms $\mathcal{F}_{\mathrm{MAH}}$ based on modified admissible hash function [13] and $\mathcal{F}_{\mathrm{AFF}}$ based on affine function. Using our generic construction in section 4.1, we can get two LPHFs with high min-entropy from both $\mathcal{F}_{\mathrm{MAH}}$ and $\mathcal{F}_{\mathrm{AFF}}$, which are denoted by $\mathcal{H}_{\mathrm{MAH}}$ and $\mathcal{H}_{\mathrm{AFF}}$ respectively.

(2) $\mathcal{H}_{\mathrm{MAH}}$: a $(1, v, \overline{m}^4 u(\ell+1), \mathsf{negl}(\lambda), \mathcal{O}(\frac{\epsilon^{\varphi}}{Q^{\varphi}}))$ LPHF with high min-entropy, where $u = \omega(\log^2 \lambda)$, v is an arbitrary polynomial in λ and $\varphi > 1$ is the constant satisfying $s = 1 - 2^{-\frac{1}{\varphi}}$, where $s \in \{0,1\}$ is the relative distance of the underlying error correcting code. We can take φ as close to 1 as one wants. In addition, the key of $\mathcal{H}_{\mathrm{MAH}}$ only consists of $u = \omega(\log^2 \lambda)$ matrices.

(3) $\mathcal{H}_{\mathrm{AFF}}$: a $(1, v, \mathsf{poly}(\lambda), \mathsf{negl}(\lambda), \mathcal{O}(\frac{\epsilon}{\ell^2 Q}))$ LPHF with high min-entropy, where v is an arbitrary polynomial in λ. Furthermore, the key of $\mathcal{H}_{\mathrm{AFF}}$ only consists of $\omega(\log \lambda)$ matrices.

As mentioned in section 4.2, the full secure IBE scheme in [19] implies an instantiation of LPHF with high min-entropy.

(4) $\mathcal{H}_{\mathrm{Yam16}}$ in Yam16 [19]: a $(1, v, \overline{m}(1 + 4\ell n^c), \mathsf{negl}(\lambda), \frac{\epsilon^2}{32(\ell+1)Q^2})$ LPHF with high min-entropy, where v is an arbitrary polynomial in λ, $c = c_1 + c_2$ and c_1, c_2 satisfy that $\frac{n^{c_1}}{2} \geq Q + 1$ and $n^{-c_2} \leq \epsilon$. Moreover, the key of $\mathcal{H}_{\mathrm{Yam16}}$ only consists of $\sqrt{\lambda}$ matrices.

Embedding these four LPHFs with high min-entropy into our generic DIBE construction, we can obtain four PRIV1-ID-INDr-secure DIBE schemes (Figure 3) in the auxiliary-input setting, under the lWE assumption.

Fig. 3. Four Adaptively Secure DIBE Schemes in the Auxiliary-Input Setting.

Schemes	# of $\mathbb{Z}_q^{n\times m}$ matrix |mpk|	Rounding Parameter p	Message Space t	Sample Width σ	Reduction Cost
$\mathrm{DIBE}_{\mathrm{MAH}}$	$\omega(\log^2 \lambda)$	$\widetilde{\mathcal{O}}(n^{6.5+5.5\eta})$	$\widetilde{\mathcal{O}}(n^{6+5\eta})$	$\widetilde{\mathcal{O}}(n^{5+4\eta})$	$\mathcal{O}(\frac{\epsilon^{\varphi+1}}{Q^{\varphi}})^{\dagger}$
$\mathrm{DIBE}_{\mathrm{AFF}}$	$\omega(\log \lambda)$	$\mathsf{poly}(n)$	$\mathsf{poly}(n)$	$\mathsf{poly}(n)$	$\mathcal{O}(\frac{\epsilon^2}{\ell^2 Q})$
$\mathrm{DIBE}_{\mathrm{ZCZ16}}$	$\mathcal{O}(\log Q)$	$\widetilde{\mathcal{O}}(n^{c+4+3\eta})$	$\widetilde{\mathcal{O}}(n^{c+3.5+2.5\eta})$	$\widetilde{\mathcal{O}}(n^{c+2.5+1.5\eta})^{\ddagger}$	$\mathcal{O}(\frac{\epsilon}{\ell Q^2})$
$\mathrm{DIBE}_{\mathrm{Yam16}}$	$\sqrt{\lambda}$	$\widetilde{\mathcal{O}}(n^{c+3.5+2.5\eta})$	$\widetilde{\mathcal{O}}(n^{c+3+2\eta})$	$\widetilde{\mathcal{O}}(n^{c+2+\eta})^{\S}$	$\mathcal{O}(\frac{\epsilon^3}{\ell Q^2})$

$|\mathrm{mpk}|, |\mathrm{ct}|$ show the size of the master public keys and ciphertexts, respectively. Q and ϵ denote the number of key extraction queries and the advantage, respectively. $\mathsf{poly}(n)$ represents a fixed but large polynomial that does not depend Q and ϵ. To measure the reduction cost, we show the advantage of the LWE algorithm constructed from the adversary against the corresponding DIBE scheme.

† $\varphi > 1$ is the constant satisfying $s = 1 - 2^{-\frac{1}{\varphi}}$, where $s \in \{0,1\}$ is the relative distance of the underlying error correcting code. We can take φ as close to 1 as one wants.

‡ c is the smallest integer satisfying that $n^c \geq Q + 1$.

§ $c = c_1 + c_2$ and c_1, c_2 are the smallest integers satisfying that $\frac{n^{c_1}}{2} \geq Q + 1$ and $n^{-c_2} \leq \epsilon$.

6 Acknowledgement

We thank the anonymous Security and Communication Networks' reviewers for their helpful comments. Yamin Liu was supported by the National Natural Science Foundation of China **61502480**. This work was also supported by the National Natural Science Foundation of China (No. 61772515, No. 61602473, No. 61572495, No.61502484) and the National Cryptography Development Fund MMJJ20170116.

References

1. S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT 2010*, pages 553–572, 2010.
2. M. Ajtai. Generating hard instances of the short basis problem. In *ICALP 1999*, pages 1–9, 1999.
3. J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. *Theory Comput. Syst.*, 48(3):535–553, 2011.
4. D. Apon, X. Fan, and F. Liu. Compact identity based encryption from lwe. *IACR Cryptology ePrint Archive*, 2016:125, 2016.
5. M. Bellare, E. Kiltz, C. Peikert, and B. Waters. Identity-based (lossy) trapdoor functions and applications. In *EUROCRYPT 2012*, pages 228–245, 2012.
6. A. Bogdanov, S. Guo, D. Masny, S. Richelson, and A. Rosen. On the hardness of learning with rounding over small modulus. In *TCC 2016-A*, pages 209–224, 2016.
7. X. Boyen and Q. Li. Towards tightly secure lattice short signature and id-based encryption. In *ASIACRYPT 2016, Part II*, pages 404–434, 2016.
8. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. *J. Cryptology*, 25(4):601–639, 2012.
9. Y. Dodis, R. Ostrovsky, L. Reyzin, and A. D. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008.
10. A. Escala, J. Herranz, B. Libert, and C. Ràfols. Identity-based lossy trapdoor functions: New definitions, hierarchical extensions, and implications. In *PKC 2014*, pages 239–256, 2014.
11. F. Fang, B. Li, X. Lu, Y. Liu, D. Jia, and H. Xue. (deterministic) hierarchical identity-based encryption from learning with rounding over small modulus. In *AsiaCCS 2016*, pages 907–912, 2016.
12. S. Goldwasser, Y. T. Kalai, C. Peikert, and V. Vaikuntanathan. Robustness of the learning with errors assumption. In *ICS 2010*, pages 230–240, 2010.
13. T. Jager. Verifiable random functions from weaker assumptions. In *TCC 2015, Part II*, pages 121–143, 2015.
14. S. Katsumata and S. Yamada. Partitioning via non-linear polynomial functions: More compact ibes from ideal lattices and bilinear maps. In *ASIACRYPT 2016*, pages 682–712, 2016.
15. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT 2012*, pages 700–718, 2012.
16. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009.
17. A. Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO 1984*, pages 47–53, 1984.

18. X. Xie, R. Xue, and R. Zhang. Deterministic public key encryption and identity-based encryption from lattices in the auxiliary-input setting. In *SCN 2012*, pages 1–18, 2012.
19. S. Yamada. Adaptively secure identity-based encryption from lattices with asymptotically shorter public parameters. In *EUROCRYPT 2016*, pages 32–62, 2016.
20. S. Yamada. Asymptotically compact adaptively secure lattice ibes and verifiable random functions via generalized partitioning techniques. In *CRYPTO 2017, Part III*, pages 161–193, 2017.
21. D. Zhang, F. Fang, B. Li, and X. Wang. Deterministic identity-based encryption from lattices with more compact public parameters. In *IWSEC 2017*, pages 215–230, 2017.
22. J. Zhang, Y. Chen, and Z. Zhang. Programmable hash functions from lattices: Short signatures and ibes with small key sizes. In *CRYPTO 2016, Part III*, pages 303–332, 2016.

A Preliminaries

Lattice Background. For positive integers q, n, m, and a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, the m-dimensional integer lattices are defined as:

$$\Lambda_q(\mathbf{A}) = \{\mathbf{y} : \mathbf{y} = \mathbf{A}^\top \mathbf{s} \text{ for some } \mathbf{s} \in \mathbb{Z}^n\} \text{ and } \Lambda_q^\perp(\mathbf{A}) = \{\mathbf{y} : \mathbf{A}\mathbf{y} = \mathbf{0} \mod q\}.$$

Let $\mathbf{S}$ be a set of vectors $\mathbf{S} = \{\mathbf{s}_1, \cdots, \mathbf{s}_n\}$ in $\mathbb{R}^m$. We use $\widetilde{\mathbf{S}} = \{\widetilde{\mathbf{s}}_1, \cdots, \widetilde{\mathbf{s}}_n\}$ to denote the Gram-Schmidt orthogonalization of the vectors $\mathbf{s}_1, \cdots, \mathbf{s}_n$ in that order, and $\|\mathbf{S}\|$ to denote the length of the longest vector in $\mathbf{S}$. For a real-valued matrix $\mathbf{R}$, let $s_1(\mathbf{R}) = \max_{\|\mathbf{u}\|=1} \|\mathbf{R}\mathbf{u}\|$ (respectively, $\|\mathbf{R}\|_\infty = \max \|\mathbf{r}_i\|_\infty$) denote the operator norm (respectively, infinity norm) of $\mathbf{R}$.

For $\mathbf{x} \in \Lambda$, define the Gaussian function $\rho_{s,\mathbf{c}}(\mathbf{x})$ over $\Lambda \subseteq \mathbb{Z}^m$ centered at $\mathbf{c} \in \mathbb{R}^m$ with parameter $s > 0$ as $\rho_{s,\mathbf{c}}(\mathbf{x}) = \exp(-\pi||\mathbf{x} - \mathbf{c}||/s^2)$. Let $\rho_{s,\mathbf{c}}(\Lambda) = \sum_{\mathbf{x}\in\Lambda} \rho_{s,\mathbf{c}}(\mathbf{x})$, and define the discrete Gaussian distribution over Λ as $\mathcal{D}_{\Lambda,s,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{s,\mathbf{c}}(\mathbf{x})}{\rho_{s,\mathbf{c}}(\Lambda)}$, where $\mathbf{x} \in \Lambda$. For simplicity, $\rho_{s,\mathbf{0}}$ and $\mathcal{D}_{\Lambda,s,\mathbf{0}}$ are abbreviated as ρ_s and $\mathcal{D}_{\Lambda,s}$, respectively.

Learning with Errors Assumption. The learning with errors (LWE) problem, denoted by $\mathrm{LWE}_{q,n,m,\alpha}$, was first proposed by Regev [16]. For integer $n, m = m(n)$, a prime integer $q > 2$, an error rate $\alpha \in (0, 1)$, the LWE problem $\mathrm{LWE}_{q,n,m,\alpha}$ is to distinguish the following pairs of distributions: $\{\mathbf{A}, \mathbf{A}^\top\mathbf{s} + \mathbf{e}\}$ and $\{\mathbf{A}, \mathbf{u}\}$, where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}, \mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, \mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$ and $\mathbf{e} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^m,\alpha q}$. Regev [16] showed that solving decisional $\mathrm{LWE}_{q,n,m,\alpha}$ (denoted by $\mathrm{DLWE}_{q,n,m,\alpha}$) for $\alpha q > 2\sqrt{2n}$ is (quantumly) as hard as approximating the SIVP and GapSVP problems to within $\widetilde{\mathcal{O}}(n/\alpha)$ factors in the worst case.

Lemma 3 ([12], Theorem 5; [18], Lemma 7). *Let $k \log t > \log q + \omega(\log(\lambda)), t = \mathsf{poly}(\lambda)$. Let $\widehat{\mathcal{M}}$ be any distribution over $\mathbb{Z}_t^n$ and $\mathcal{H}_{\mathrm{hard}}$ be the class of all functions $h : \mathbb{Z}_t^n \to \{0,1\}^*$ that are $2^{-k\log(t)}$ hard to invert with respect to the distribution $\widehat{\mathcal{M}}$. For any super-polynomial $q = q(\lambda)$, any $m = \mathsf{poly}(n)$, and*

any $\alpha, \theta \in (0,1)$ *such that* $\alpha/\theta = \mathsf{negl}(\lambda)$*, then the following pairs of distributions:* $\big(\mathbf{A}, \mathbf{A}^\top \mathbf{s} + \mathbf{e}, h(\mathbf{s})\big)$ *and* $(\mathbf{A}, \mathbf{u}, h(\mathbf{s}))$ *are hard to distinguish, where* $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}, \mathbf{s} \xleftarrow{\$} \widehat{\mathcal{M}} \subseteq \mathbb{Z}_t^n, \mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$ *and* $\mathbf{e} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^m,\theta q}$*. Assuming the (standard)* $\mathrm{LWE}_{q,z,m,\alpha}$ *assumption, where* $z \stackrel{\triangle}{=} \frac{k\log(t)-\omega(\log(\lambda))}{\log(q)}$.

For simplicity, we use $\mathrm{DLWE}_{q,n,m,\beta,\mathcal{H}_{\mathrm{hard}}}$ to denote the problem of distinguishing the above two distributions: $\big(\mathbf{A}, \mathbf{A}^\top \mathbf{s} + \mathbf{e}, h(\mathbf{s})\big)$ and $(\mathbf{A}, \mathbf{u}, h(\mathbf{s}))$. According to Lemma 3, assuming the $\mathrm{DLWE}_{q,z,m,\alpha}$, then the $\mathrm{DLWE}_{q,n,m,\beta,\mathcal{H}_{\mathrm{hard}}}$ problem is also intractable, where $z \stackrel{\triangle}{=} \frac{k\log(t)-\omega(\log(\lambda))}{\log(q)}$. In the following, we describe some useful facts that will be used in our generic DIBE construction.

Gadget Matrix. As mentioned by [15], for $m > n\lceil \log q \rceil$, there exists a full-rank matrix $\mathbf{G} \in \mathbb{Z}_q^{n\times m}$ such that the lattice $\Lambda_q^\perp(\mathbf{G})$ has a public known basis $\mathbf{T_G} \in \mathbb{Z}_q^{m\times m}$ with $\|\widetilde{\mathbf{T_A}}\| \leq \sqrt{5}$. Moreover, there exists a deterministic PPT algorithm $\mathbf{G}^{-1}$ which takes the input $\mathbf{U} \in \mathbb{Z}_q^{n\times m}$ and outputs $\mathbf{V} = \mathbf{G}^{-1}(\mathbf{U})$ such that $\mathbf{V} \in \{0,1\}^{m\times m}$ and $\mathbf{GV} = \mathbf{U}$.

Lemma 4. *Let* $p, q, n, \overline{m}$ *be positive integers with* $q \geq p \geq 2$ *and* q *prime. There exists PPT algorithms such that*

- *([2, 3]):* $\mathsf{TrapGen}(1^n, 1^{\overline{m}}, q)$ *a randomized algorithm that, when* $\overline{m} \geq 6n\lceil \log q \rceil$*, outputs a pair* $(\mathbf{A}, \mathbf{T_A}) \in \mathbb{Z}_q^{n\times\overline{m}} \times \mathbb{Z}^{\overline{m}\times\overline{m}}$ *such that* $\mathbf{A}$ *is statistically close to uniform in* $\mathbb{Z}_q^{n\times\overline{m}}$ *and* $\mathbf{T_A}$ *is a basis of* $\Lambda_q^\perp(\mathbf{A})$*, satisfying* $\|\widetilde{\mathbf{T_A}}\| \leq \mathcal{O}(\sqrt{n\log q})$ *with overwhelming probability.*
- *([8]):* $\mathsf{SampleBasisLeft}(\mathbf{A}, \mathbf{B}, \mathbf{T_A}, \sigma)$ *a randomized algorithm that, given a full rank matrix* $\mathbf{A} \in \mathbb{Z}_q^{n\times\overline{m}}$*, a matrix* $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$*, a basis* $\mathbf{T_A}$ *of* $\Lambda_q^\perp(\mathbf{A})$*, a parameter* $\sigma \geq \|\widetilde{\mathbf{T_A}}\| \cdot \omega(\sqrt{\log(\overline{m}+m)})$*, then outputs a basis* $\mathbf{T_F}$ *of* $\Lambda_q^\perp(\mathbf{F})$ *for* $\mathbf{F} = [\mathbf{A}|\mathbf{B}]$ *with* $\|\mathbf{T_F}\| \leq \mathcal{O}(\sigma \cdot m)$*.*
- *([1]):* $\mathsf{SampleBasisRight}(\mathbf{A}, \mathbf{G}, \mathbf{R}, \mathbf{S}, \mathbf{u}, \mathbf{T_G}, \sigma)$ *a randomized algorithm that, given a full rank matrix* $\mathbf{A} \in \mathbb{Z}_q^{n\times\overline{m}}$*, a matrix* $\mathbf{R} \in \mathbb{Z}_q^{\overline{m}\times m}$*, an invertible matrix* $\mathbf{S} \in \mathbb{Z}_q^{n\times n}$*, a vector* $\mathbf{u} \in \mathbb{Z}_q^n$ *and* $\sigma \geq \|\widetilde{\mathbf{T_G}}\| \cdot s_1(\mathbf{R}) \cdot \omega(\sqrt{\log m})$*, then it outputs a basis* $\mathbf{T_F}$ *of* $\Lambda_q^\perp(\mathbf{F})$ *for* $\mathbf{F} = [\mathbf{A}|\mathbf{AR}+\mathbf{SG}]$ *with* $\|\mathbf{T_F}\| \leq \mathcal{O}(\sigma \cdot m)$*.*
- *([11]):* $\mathsf{Invert}(\mathbf{c}, \mathbf{A}, \mathbf{T_A})$ *that, given a full rank matrix* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$*, a basis* $\mathbf{T_A}$ *of* $\Lambda_q^\perp(\mathbf{A})$ *with* $\|\mathbf{T_A}\| < p/(2\sqrt{m})$*, and* $\mathbf{c} = \lfloor \mathbf{A}^\top \mathbf{m} \rceil_p$ *outputs* $\mathbf{m}$*, where* $\mathbf{m} \in \mathbb{Z}_t^n$ *with* $t \leq q$*.*
- *(Generalized Leftover Hash Lemma [1, 9]): For* $\overline{m} > (n+1)\log q + \omega(\log n)$ *and prime* $q > 2$*, let* $\mathbf{R} \xleftarrow{\$} \{-1,1\}^{\overline{m}\times k}$ *and* $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times\overline{m}}, \mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{n\times k}$ *be uniformly random matrices. Then the distribution* $(\mathbf{A}, \mathbf{AR}, \mathbf{R}^\top \mathbf{w})$ *is* $\mathsf{negl}(n)$*-close to the distribution* $(\mathbf{A}, \mathbf{B}, \mathbf{R}^\top \mathbf{w})$ *for all vector* $\mathbf{w} \in \mathbb{Z}_q^{\overline{m}}$*. When* $\mathbf{w}$ *is always* $\mathbf{0}$*, this lemma is called Leftover Hash Lemma.*

In [14], Katsuamta and Yamada introduced the "Noise Rerandomization" lemma which plays an important role in the security proof because of creating a well distributed challenge ciphertext.

Lemma 5 (Noise Rerandomization [14]). *Let q, w, m be positive integers and r a positive real number with $r > \max\{\omega(\sqrt{\log m}), \omega(\sqrt{\log w})\}$. For arbitrary column vector $\mathbf{b} \in \mathbb{Z}_q^m$, vector $\mathbf{e}$ chosen from $\mathcal{D}_{\mathbb{Z}^m,r}$, any matrix $\mathbf{V} \in \mathbb{Z}^{w\times m}$ and positive real number $\sigma > s_1(\mathbf{V})$, there exists a PPT algorithm $\mathsf{ReRand}(\mathbf{V}, \mathbf{b} + \mathbf{e}, r, \sigma)$ that outputs $\mathbf{b}' = \mathbf{V}\mathbf{b} + \mathbf{e}' \in \mathbb{Z}^w$ where $\mathbf{e}'$ is distributed statistically close to $\mathcal{D}_{\mathbb{Z}^w,2r\sigma}$.*

Partitioning Functions with Compatible Algorithms. In [20], Yamada defined the notion of partitioning functions by slightly generalizing the balanced admissible hash function [13] and used this notion to construct compact adaptively secure lattice IBE schemes. Furthermore, in order to construct IBE from lattices, the underlying partitioning function should be compatible with the structure of lattices.

Definition 4 ([20]). *Let $\mathcal{F} = \{\mathcal{F}_\lambda : \mathcal{K}_\lambda \times \mathrm{ID}_\lambda \to \{0,1\}\}$ be an ensemble of function families. We say that $\mathcal{F}$ is a partitioning function, if there exists an efficient algorithm $\mathsf{PrtSmp}(1^\lambda, Q, \epsilon)$, which takes as input polynomially bounded $Q = Q(\lambda) \in \mathbb{N}$ and noticeable $\epsilon = \epsilon(\lambda) \in (0, 1/2]$ and outputs $\mathcal{K}$ such that:*

1. *There exists $\lambda_0 \in \mathbb{N}$ such that $\Pr\left[K \in \mathcal{K}_\lambda : K \xleftarrow{\$} \mathsf{PrtSmp}(1^\lambda, Q, \epsilon)\right] = 1$ for all $\lambda > \lambda_0$. Here, λ_0 may depend on functions $Q(\lambda)$ and $\epsilon(\lambda)$.*
2. *For $\lambda > \lambda_0$, there exists $\gamma_{\max}(\lambda)$ and $\gamma_{\min}(\lambda)$ that depend on $Q(\lambda)$ and $\epsilon(\lambda)$ such that for all $id_1, \cdots, id_Q, id^*$ with $id^* \notin \{id_1, \cdots, id_Q\}$, the following holds*

$$\gamma_{\max}(\lambda) \geq \Pr[\mathcal{F}(K, id_1) = \cdots \mathcal{F}(K, id_Q) = 1 \wedge \mathcal{F}(K, id^*) = 0] \geq \gamma_{\min}(\lambda).$$

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

The deterministic algorithms $(\mathsf{Encode}, \mathsf{PubEval}, \mathsf{TrapEval})$ are called δ_{PF}-compatible with a function family $\{\mathcal{F}_\lambda : \mathcal{K} \times \mathrm{ID} \to \{0,1\}\}$ if they are efficient and satisfy the following properties:

- $\mathsf{Encode}(K \in \mathcal{K}) \to k \in \{0,1\}^u$.
- $\mathsf{PubEval}(id \in \mathrm{ID}, \{\mathbf{B}_i \in \mathbb{Z}_q^{n\times m}\}_{i\in[u]}) \to \mathbf{B}_{id} \in \mathbb{Z}_q^{n\times m}$.
- $\mathsf{TrapEval}(K \in \mathcal{K}, id \in \mathrm{ID}, \mathbf{A} \in \mathbb{Z}_q^{n\times m}, \{\mathbf{R}_i \in \mathbb{Z}_q^{m\times m}\}_{i\in[u]}) \to \mathbf{R}_{id} \in \mathbb{Z}_q^{m\times m}$. We require that the following holds:

$$\mathsf{PubEval}(id, \{\mathbf{A}\mathbf{R}_i + k_i\mathbf{G}\}_{i\in[u]}) = \mathbf{A}\mathbf{R}_{id} + \mathcal{F}(K, id) \cdot \mathbf{G},$$

where k_i is the i-th bit of $k = \mathsf{Encode}(K \in \mathcal{K}) \in \{0,1\}^u$. Furthermore, if $\mathbf{R}_i \in \{-1, 0, 1\}^{m\times m}$ for all $i \in [u]$, we have $\|\mathbf{R}_{id}\|_\infty \leq \delta_{\mathrm{PF}}$.