SCIENCE CHINA
Information Sciences

• **Supplementary File** •

A More Compact Multi-id Identity-based FHE Scheme in the Standard Model and Its Applications

Xueqing WANG[1,2*], Biao WANG[1,2], Bei LIANG[3] & Rui XUE[1,2*]

[1]*State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing* 100093*, China;*
[2]*School of Cyber Security, University of Chinese Academy of Sciences, Beijing* 100049*, China;*
[3]*Chalmers University of Technology, Gothenburg, Sweden*

Appendix A The Simpler Ciphertext Expansion Technique

According to the two types of identity mapping in existed IBE schemes: one type is GPV-IBE [1] and another is other lattice-based IBE schemes [2–7], there are two types of vector ciphertexts in original IBE schemes and hence two types of matrix ciphertexts in dual-Regev encryption form of [8,9]. The ciphertext expansion technique in identity case is to expand the matrix ciphertext under a secret key $\mathbf{s}_{id_i} = (1, \mathbf{w}_{id_i}^T)^T$

$$\textit{Type I. } \mathbf{C}_{id_i} := \mathbf{S}_i(\mathbf{u}_{id_i} || -\mathbf{A}) + \mathbf{E}_i + \mu_i \mathbf{G}_{m+1}, \text{ the case of [1]},$$

where $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, $\mathbb{Z}_q^n \ni \mathbf{u}_{id_i} := \mathbf{A}\mathbf{w}_{id_i}, \mathbf{S}_i \in \mathbb{Z}_q^{(m+1)\ell_q \times n}, \mathbf{E}_i \in \mathbb{Z}^{(m+1)\ell_q \times (m+1)}, \mathbf{G}_{m+1} := \mathbf{I}_{m+1} \otimes \mathbf{g}^T, \mathbf{g}^T := (2^0, 2^1, \ldots, 2^{\ell_q - 1}), \ell_q := \lceil \log_2 q \rceil$, or

$$\textit{Type II. } \mathbf{C}_{id_i} := \mathbf{S}_i(\mathbf{u} || -\mathbf{A}_{id_i}) + \mathbf{E}_i + \mu_i \mathbf{G}_{m+1}, \text{ the case of [2–7]},$$

where $\mathbb{Z}_q^n \ni \mathbf{u} := \mathbf{A}_{id_i}\mathbf{w}_{id_i}$, satisfying $\mathbf{C}_{id_i}\mathbf{s}_{id_i} \approx \mu_i \mathbf{G}_{m+1}\mathbf{s}_{id_i}$ according to the approximate eigenvector method [8], into a matrix ciphertext under a set of secret keys $\{\mathbf{s}_{id_i}\}_{i\in[N]}$ for N involving identities in a homomorphic evaluation

$$\widehat{\mathbf{C}}_{id_i} = \begin{array}{c} \\ 1 \\ \vdots \\ i \\ \vdots \\ N \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & \cdots & i & \cdots & N \end{array} \\ \begin{pmatrix} \mathbf{C}_{id_i} & \cdots & \mathbf{X}_1 & & \\ & \ddots & \vdots & & \\ & & \mathbf{C}_{id_i} & & \\ & & \vdots & \ddots & \\ & & \mathbf{X}_N & \cdots & \mathbf{C}_{id_i} \end{pmatrix} \end{array},$$

satisfying

$$\widehat{\mathbf{C}}_{id_i} \cdot \begin{pmatrix} \mathbf{s}_{id_1} \\ \vdots \\ \mathbf{s}_{id_i} \\ \vdots \\ \mathbf{s}_{id_N} \end{pmatrix} \approx \mu_i \mathbf{G}_{m+1} \begin{pmatrix} \mathbf{s}_{id_1} \\ \vdots \\ \mathbf{s}_{id_i} \\ \vdots \\ \mathbf{s}_{id_N} \end{pmatrix},$$

for $j \in [N] \setminus \{i\}$,

$$\textit{Type I. } \mathbf{C}_{id_i}\mathbf{s}_{id_j} \approx \mathbf{S}_i(\mathbf{u}_{id_i} - \mathbf{u}_{id_j}) + \mu_i \mathbf{G}_{m+1}\mathbf{s}_{id_j},$$

or

$$\textit{Type II. } \mathbf{C}_{id_i}\mathbf{s}_{id_j} \approx \mathbf{S}_i(\mathbf{u} - \mathbf{A}_{id_i}\mathbf{w}_{id_j}) + \mu_i \mathbf{G}_{m+1}\mathbf{s}_{id_j},$$

the term $\mathbf{X}_j\mathbf{s}_{id_i}$ is used to eliminate the residue in $\mathbf{C}_{id_i}\mathbf{s}_{id_j}$, that is,

$$\textit{Type I. } \mathbf{X}_j\mathbf{s}_{id_i} \approx \mathbf{S}_i(\mathbf{u}_{id_j} - \mathbf{u}_{id_i})$$

* Corresponding author (email: wangxueqing@iie.ac.cn, xuerui@iie.ac.cn)

or

$$\textit{Type II. } \mathbf{X}_j\mathbf{s}_{id_i} \approx \mathbf{S}_i(\mathbf{A}_{id_i}\mathbf{w}_{id_j} - \mathbf{u}),$$

where $\mathbf{X}_j$ can be viewed as a pseudo encryption of $\mathbf{S}_i(\mathbf{u}_{id_j} - \mathbf{u}_{id_i})$ or $\mathbf{S}_i(\mathbf{A}_{id_i}\mathbf{w}_{id_j} - \mathbf{u})$. And we will use the technique of homomorphic linear combination, similarly as that in [10], to compute $\mathbf{X}_j$ from an encryption of $\mathbf{S}_i$ and another public and deterministic term. However, for *Type I*, only GPV-IBE meets the condition of public and deterministic $\mathbf{u}_{id_j} - \mathbf{u}_{id_i}$ due to the use of RO. And for *Type II*, the condition of public and deterministic $\mathbf{A}_{id_i}\mathbf{w}_{id_j}$ is as far not satisfied.

Appendix B Preliminaries

Appendix B.1 Learning with Errors (LWE)

The LWE problem, introduced by Regev [11], has two variants: search variant and decision variant. In generally, the search variant of a problem is harder than its decision variant. However, the two variants of the LWE problem are equivalent with proper parameters. So we just present the decisional LWE (DLWE) problem as follows.

Definition 1 (DLWE [11])**.** For security parameter λ, let $n = n(\lambda)$, $q = q(\lambda) \geqslant 2$ be two integers, and $\chi = \chi(\lambda)$ be a distribution over $\mathbb{Z}$. The decisional learning with errors problem, denoted by $\mathsf{DLWE}_{n,q,\chi}$, is to distinguish the following two distributions:

$$U(\mathbb{Z}_q^{n+1}) := \Big\{(\mathbf{a}_i, b_i)\Big\}_{\mathbf{a}_i \xleftarrow{\$} \mathbb{Z}_q^n,\, b_i \xleftarrow{\$} \mathbb{Z}_q} \text{ and } A_{\mathbf{s},\chi} := \Big\{(\mathbf{a}_i, \langle \mathbf{a}_i, \mathbf{s}\rangle + e_i)\Big\}_{\mathbf{a}_i \xleftarrow{\$} \mathbb{Z}_q^n,\, e_i \leftarrow \chi}.$$

The $\mathsf{DLWE}_{n,q,\chi}$ assumption is that the $\mathsf{DLWE}_{n,q,\chi}$ problem is infeasible.

The $\mathsf{DLWE}_{n,q,\chi}$ assumption is standard since there are known quantum [11] and classical [12] reductions between $\mathsf{DLWE}_{n,q,\chi}$ and approximating short vector problems on lattices. In particular, these reductions take χ to be the Gaussian distribution, which is statistically indistinguishable from B-bounded distribution, for an appropriate B. We cite the definition of B-bounded distributions from [8].

Definition 2 (B-Bounded Distributions [8])**.** A distribution ensemble $\{\chi_n\}_{n\in\mathbb{N}}$, supported over the integers, is called B-bounded for any $B > 0$ if

$$\Pr_{e\leftarrow\chi_n}[|e| > B] = \mathrm{negl}(n).$$

Appendix B.2 Multi-id Identity-based Fully Homomorphic Encryption

Formally, an MIBFHE scheme is a tuple of PPT algorithms as follows.

- $\mathsf{Setup}(\cdot)$: On input a security parameter 1^λ, output public parameters $\mathtt{mpk}$ and a master secret key $\mathtt{msk}$.
- $\mathsf{KeyGen}(\cdot,\cdot)$: On input the master secret key $\mathtt{msk}$ and an identity id, derive and output a secret key sk_{id} for id.
- $\mathsf{Enc}(\cdot,\cdot,\cdot)$: On input public parameters $\mathtt{mpk}$, an identity id and a message μ, output a ciphertext c that encrypts μ under identity id. Note that c implicitly contains id.
- $\mathsf{Dec}(\cdot,\cdot)$: On input N secret keys $(sk_{id_1}, \ldots, sk_{id_N})$ for $id_1, \ldots, id_N$ respectively and a ciphertext $\hat{c}$, output μ if $\hat{c}$ is a valid ciphertext under identities $id_1, \ldots, id_N$, and output a failure symbol $\perp$ otherwise.
- $\mathsf{Eval}(\cdot,\cdot,\cdot)$: On input public parameters $\mathtt{mpk}$, a circuit f and ciphertexts $(c_1, \ldots, c_t)$ possibly under different identities, output an evaluated ciphertext $\hat{c}$.

Some properties for an MIBFHE scheme are defined as follows.

- **Fully dynamic property (The identity variant of that from [13]).** Let $N = N(\lambda)$, $t = t(\lambda)$ be any polynomial on the security parameter λ. For any f in the family of admissible circuits, for any $(id_1, \ldots, id_N)$, $(\hat{c}_1, \ldots, \hat{c}_t)$ such that $\mathsf{Dec}((sk_{id_{j_1}}, \ldots, sk_{id_{j_{s_j}}}), \hat{c}_j) = \mu_j$, where $j \in [t]$, $s_j \in [N]$, $\{sk_{id_{j_1}}, \ldots, sk_{id_{j_{s_j}}}\} \subseteq \{sk_{id_1}, \ldots, sk_{id_N}\}$, the following holds:

$$\Pr\left[\mathsf{Dec}((sk_{id_1}, \ldots, sk_{id_N}), \mathsf{Eval}(\mathtt{mpk}, f, (\hat{c}_1, \ldots, \hat{c}_t))) \neq f(\mu_1, \ldots, \mu_t)\right] = \mathrm{negl}(\lambda),$$

where $(\mathtt{mpk}, \mathtt{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$, $sk_{id_i} \leftarrow \mathsf{KeyGen}(\mathtt{msk}, id_i)$ for $i \in [N]$. Otherwise, the scheme, in which the result ciphertext from some homomorphic evaluation cannot be further computed while new identities joining in, is called single-hop.

- **Compactness.** The size of the result ciphertext from homomorphic evaluation is independent of the size or the depth of the input circuit, or the number of input ciphertext N_c or the number of different involving identities N_i. Note that the size in our scheme is at least independent of N_c, better than that in [14] depends on N_c.

- **Security.** In this work, we focus on IND-sID-CPA security, which is the same as that for IBE except that here the adversary has an additional homomorphic evaluation ability. In the following, we present its definition as a game interactively executed between a challenger $\mathcal{C}$ and a PPT adversary $\mathcal{A}$.

- **Init Phase.** $\mathcal{A}$ submits a target identity id^*.

- **Setup Phase.** $\mathcal{C}$ calls $\mathsf{Setup}(1^\lambda)$ to generate $(\mathtt{mpk}, \mathtt{msk})$ and sends $\mathtt{mpk}$ to $\mathcal{A}$.

- **Secret-Key Query Phase I.** In this phase, $\mathcal{A}$ can issue any polynomial number of queries. When $\mathcal{A}$ asks for a secret key on an identity id, $\mathcal{C}$ checks whether $id = id^*$: if yes, returns $\perp$; otherwise, it generates and returns $sk_{id} \leftarrow \mathsf{KeyGen}(\mathtt{msk}, id)$.

- **Challenge Phase.** On receiving $\mathcal{A}$'s messages μ, $\mathcal{C}$ picks $\sigma \in \{0,1\}$ uniformly at random and generates $\mathbf{c}_{id^*} \leftarrow \mathsf{Enc}(\mathtt{mpk}, id^*, \mu)$. And then $\mathcal{C}$ sends $\mathbf{c}_{id^*}$ to $\mathcal{A}$.

- **Secret-Key Query Phase II.** This is the same as Secret-Key Query Phase I.

- **Guess Phase.** $\mathcal{A}$ outputs its guess $\sigma' \in \{0,1\}$.

Note that since IBE is a particular case of MIBFHE with Dec in a single identity case and without the algorithm Eval, whose correctness can be easily defined and security definition is the same as that of MIBFHE, we omit the formal definition of IBE. In this work, we consider INDr-sID-CPA security of IBE, originated from [2], which implies anonymity and IND-sID-CPA security.

Appendix B.3 Indistinguishability Obfuscation

We present the formal definition following the syntax of Garg et al. [15].

Definition 3 (Indistinguishability Obfuscator($i\mathcal{O}$) [15])**.** A uniform PPT machine $i\mathcal{O}$ is called an indistinguishability obfuscator for a circuit class $\{\mathcal{C}_\lambda\}$ if the following holds:

- **(Correctness)**. For all security parameters $\lambda \in \mathbb{N}$, all $C \in \mathcal{C}_\lambda$ and all inputs x, we have that

$$\Pr[C'(x) = C(x) : C' \leftarrow i\mathcal{O}(\lambda, C)] = 1.$$

- **(Indistinguishability)**. For any (not necessarily uniform) PPT distinguisher $(\mathsf{Samp}, \mathcal{D})$, there exists a negligible function negl such that the following holds: if $\Pr[\forall x, \mathcal{C}_0(x) = \mathcal{C}_1(x) : (C_0, C_1, \sigma) \leftarrow \mathsf{Samp}(1^\lambda)] > 1 - \mathrm{negl}(\lambda)$, then we have:

$$\begin{aligned}|\Pr[\mathcal{D}(\sigma, i\mathcal{O}(\lambda, C_0)) = 1 : (C_0, C_1, \sigma) \leftarrow \mathsf{Samp}(1^\lambda)] \\ - \Pr[\mathcal{D}(\sigma, i\mathcal{O}(\lambda, C_1)) = 1 : (C_0, C_1, \sigma) \leftarrow \mathsf{Samp}(1^\lambda)]| \leqslant \mathrm{negl}(\lambda).\end{aligned}$$

Recently, Lin and Vaikuntanathan [16] showed how to build $i\mathcal{O}$ from constant-degree multilinear maps, which was analyzed to be with degree requirement in excess of 30. Then Ananth and Sahai [17] improved the result of yielding an $i\mathcal{O}$ construction from degree-5 multilinear maps, through leveraging a variant of functional encryption for degree-5 arithmetic circuits.

Appendix B.4 Puncturable Pseudorandom Functions

Below we present a slight variant of the definition from [18]:

Definition 4 (Puncturable Pseudorandom Functions (PPRF) (A slight variant of the definition from [18]))**.** A puncturable family of PRFs $\mathsf{F} : \mathcal{K} \times \{0,1\}^k \to \{0,1\}^m$ is equipped with two algorithms $\mathsf{Key}, \mathsf{Puncture}$, a key space $\mathcal{K}$ and a pair of computable functions $k(\cdot)$ and $m(\cdot)$, satisfying the following conditions:

- **(Functionality preserved under puncturing)**. For every PPT adversary $\mathcal{A}$ such that $\mathcal{A}(1^\lambda)$ outputs a set $S \subseteq \{0,1\}^{k(\lambda)}$, then for all $x \in \{0,1\}^{k(\lambda)}$ where $x \notin S$, we have that:

$$\Pr[\mathsf{F}(K, x) = \mathsf{F}(K(S), x) : K \leftarrow \mathsf{Key}(1^\lambda), K(S) \leftarrow \mathsf{Puncture}(K, S)] = 1.$$

- **(Pseudorandomness at punctured points)**. For every PPT adversary $(\mathcal{A}_1, \mathcal{A}_2)$ such that $\mathcal{A}_1(1^\lambda)$ outputs a set $S \subseteq \{0,1\}^{k(\lambda)}$ and $x \in S$, consider an experiment where $K \leftarrow \mathsf{Key}(1^\lambda)$ and $K(S) \leftarrow \mathsf{Puncture}(K, S)$. Then we have

$$|\Pr[\mathcal{A}_2(K(S), x, \mathsf{F}(K, x)) = 1] - \Pr[\mathcal{A}_2(K(S), x, y \xleftarrow{\$} \{0,1\}^m) = 1]| \leqslant \mathrm{negl}(\lambda).$$

Appendix C Proof of Theorem 1

To prove the theorem, we consider the following game sequence, in which, apart from the first one, each game is deduced and described by modifications to its previous one. Suppose there exists a PPT adversary $\mathcal{A}$ breaking our scheme's INDr-sID-CPA security with non-negligible advantage.

Game 0. This is just the original INDr-sID-CPA security game between a PPT adversary $\mathcal{A}$ against our scheme and a challenger $\mathcal{C}$.

- **Init Phase.** $\mathcal{A}$ submits a target identity id^*.

- **Setup Phase.** $\mathcal{C}$ calls $\mathsf{IBE.Setup}(1^\lambda)$ to generate $\mathtt{mpk} = (\mathbf{A}, \mathsf{H})$, $\mathtt{msk} = K$ and sends $\mathtt{mpk}$ to $\mathcal{A}$.

- **Secret-Key Query Phase I.** In this phase, $\mathcal{A}$ can issue any polynomial number of queries. When $\mathcal{A}$ asks for a secret key on an identity id, $\mathcal{C}$ checks whether $id = id^*$: if yes, returns $\perp$; otherwise, generates and returns $\mathbf{s}_{id} \leftarrow \mathsf{IBE.KeyGen}(\mathtt{msk}, id)$.

- **Challenge Phase.** On receiving $\mathcal{A}$'s choice $\mu \in \{0,1\}$, $\mathcal{C}$ picks $\sigma \in \{0,1\}$ uniformly at random: If $\sigma = 0$, it generates $\mathbf{c}_{id^*} \leftarrow \mathsf{IBE.Enc}(\mathtt{mpk}, id^*, \mu)$; else, it chooses a random element from $\mathbb{Z}_q^{m+1}$ as the challenge ciphertext $\mathbf{c}_{id^*}$. And then $\mathcal{C}$ sends $\mathbf{c}_{id^*}$ to $\mathcal{A}$.

- **Secret-Key Query Phase II.** This is the same as Secret-Key Query Phase I.

- **Guess Phase.** $\mathcal{A}$ outputs $\sigma' \in \{0,1\}$ for guessing whether $\mathbf{c}_{id^*}$ is a ciphertext of μ or a random ciphertext.

Input: $id \in \{0,1\}^k$
Constants: $\mathbf{A}, K(\{id^*\}), id^*, \mathbf{u}_{id^*}$

1. If $id = id^*$, output $\mathbf{u}_{id^*}$.
2. $\mathbf{w}_{id} := \mathsf{F}(K(\{id^*\}), id) \in \{0,1\}^m$,
3. output $\mathbf{u}_{id} := \mathbf{A}\mathbf{w}_{id} \in \mathbb{Z}_q^n$.

Figure C1 Program RelateID'

Game 1. This game is the same as Game 0 except that the point and the manner in which $\mathbf{u}_{id^*}$ is generated. In Game 0, $\mathbf{u}_{id^*}$ is generated by the obfuscated program H in the Challenge Phase, while in this game, $\mathcal{C}$ generates it using K directly just after the PRF key generation in the Setup Phase:

$$\mathbf{w}_{id^*} := \mathsf{F}(K, id^*),\ \mathbf{u}_{id^*} := \mathbf{A}\mathbf{w}_{id^*}.$$

Game 2. This game is the same as Game 1 except that $\mathcal{C}$: (i) additionally generates $K(\{id^*\}) \leftarrow \mathsf{F.Puncture}(K, \{id^*\})$, (ii) replaces the program RelateID with RelateID' in Figure C1, and (iii) exploits $K(\{id^*\})$ instead of K to generate secret keys for $\mathcal{A}$'s queries on id:

$$\mathbf{w}_{id} := \mathsf{F}(K(\{id^*\}), id),\ \mathbf{s}_{id} = (1, \mathbf{w}_{id}^T)^T.$$

Note that the size of RelateID' is padded to be the maximum of itself and RelateID in Figure 1.

Game 3. This game is the same as Game 2 except that $\mathcal{C}$ generates $\mathbf{w}_{id^*} \xleftarrow{\$} \{0,1\}^m$ in this game instead of $\mathbf{w}_{id^*} := \mathsf{F}(K, id^*)$.

Game 4. This game is the same as Game 3 except that $\mathcal{C}$ replaces $\mathbf{u}_{id^*} := \mathbf{A}\mathbf{w}_{id^*}$ with $\mathbf{u}_{id^*} \xleftarrow{\$} \mathbb{Z}_q^n$, and therefore $\mathbf{w}_{id^*} \xleftarrow{\$} \{0,1\}^m$ can be eliminated.

Game 5. This game is identical to Game 4 except that the challenge ciphertext $\mathbf{c}_{id^*}$ is chosen as a random element in $\mathbb{Z}_q^{m+1}$.

Obviously, Game 0 and Game 1 are identical.

Lemma 1. Suppose that $i\mathcal{O}$ is an indistinguishability obfuscator, then Game 1 and Game 2 are computationally indistinguishable.

Proof. Note that, since for $id \neq id^*$, $\mathsf{F}(K, id) = \mathsf{F}(K(\{id^*\}), id)$ by the functionality preserved under puncturing in Definition 4, the only difference between Game 1 and Game 2 is that in the former game, we generate the obfuscated program by $\mathsf{H} \leftarrow i\mathcal{O}(\lambda, \mathsf{RelateID})$, while in the latter game, we obtain $\mathsf{H} \leftarrow i\mathcal{O}(\lambda, \mathsf{RelateID'})$. It's easy to see that RelateID and RelateID' have identical input-output behaviour, that is, for any input $id \in \{0,1\}^k$, the outputs of the two programs on id have the identical distribution. And then by the security of indistinguishability obfuscator in Definition 3, we have that $i\mathcal{O}(\lambda, \mathsf{RelateID})$ and $i\mathcal{O}(\lambda, \mathsf{RelateID'})$ are computationally indistinguishable. Therefore, these two games are computationally indistinguishable.

Lemma 2. Suppose that $\mathsf{F} : \mathcal{K} \times \{0,1\}^{k(\lambda)} \to \{0,1\}^m$ is a puncturable family of PRFs equipped with ($\mathsf{F.Key}$, $\mathsf{F.Puncture}$), then Game 2 and Game 3 are computationally indistinguishable.

Proof. Recall that the only difference between Game 2 and Game 3 is the manner in which $\mathbf{w}_{id^*}$ is generated. According to the property of pseudorandomness at punctured points in Definition 4, the distributions of $\mathbf{w}_{id^*} := \mathsf{F}(K, id^*)$ and $\mathbf{w}_{id^*} \xleftarrow{\$} \{0,1\}^m$ are computational indistinguishable even when the adversary knows $K(\{id^*\}), id^*$, which implies that Game 2 and Game 3 are computationally indistinguishable.

Since $\mathbf{w}_{id^*}$ in Game 3 is distributed uniformly over $\{0,1\}^m$, we have that the distributions of $\mathbf{u}_{id^*} := \mathbf{A}\mathbf{w}_{id^*}$ and $\mathbf{u}_{id^*} \xleftarrow{\$} \mathbb{Z}_q^n$ are statistically indistinguishable, by Lemma 5.1 in [1]. So Game 3 and Game 4 are statistically indistinguishable.

Lemma 3. Suppose that the $\mathsf{DLWE}_{n,q,\chi}$ assumption holds, then Game 4 and Game 5 are computationally indistinguishable.

Proof. We construct an adversary $\mathcal{B}$ against the $\mathsf{DLWE}_{n,q,\chi}$ problem by simulating the view of $\mathcal{A}$:

1. On access to an oracle which is either the uniform distribution $U(\mathbb{Z}_q^{n+1})$ or $A_{\mathbf{s},\chi}$ defined in Definition 1, and on input $\mathcal{A}$'s target identity id^*, $\mathcal{B}$ can get $m+1$ samples $(\mathbf{a}_0, b_0), (\mathbf{a}_1, b_1), \ldots, (\mathbf{a}_m, b_m)$, and it exploits these samples to set $\mathbf{A} := (\mathbf{a}_1, \ldots, \mathbf{a}_m)$, $\mathbf{u}_{id^*} := \mathbf{a}_0$, runs $K \leftarrow \mathsf{F.Key}(1^\lambda)$, $K(\{id^*\}) \leftarrow \mathsf{F.Puncture}(K, \{id^*\})$, and uses $\mathbf{A}, K(\{id^*\}), id^*, \mathbf{u}_{id^*}$ to construct program RelateID' and its obfuscated program $\mathsf{H} \leftarrow i\mathcal{O}(\lambda, \mathsf{RelateID'})$. Then, $\mathcal{B}$ sends $\mathtt{mpk} = (\mathbf{A}, \mathsf{H})$ to $\mathcal{A}$.
2. When $\mathcal{A}$ asks for a secret key on the identity id, $\mathcal{B}$ checks whether $id = id^*$: if yes, returns $\perp$; else, uses $K(\{id^*\})$ to generate $\mathbf{s}_{id}$ and returns it to $\mathcal{A}$.
3. In the Challenge Phase, on receiving $\mathcal{A}$'s choice $\mu \in \{0,1\}$, $\mathcal{B}$ picks $\sigma \in \{0,1\}$ uniformly at random: If $\sigma = 0$, it sets $\mathbf{c}_{id^*} := (b_0, -b_1, \ldots, -b_m)^T + (\mu \lfloor \frac{q}{2} \rfloor, \mathbf{0}_{1\times m})^T$; else, it chooses a random element from $\mathbb{Z}_q^{m+1}$ as the challenge ciphertext $\mathbf{c}_{id^*}$. And then it sends $\mathbf{c}_{id^*}$ to $\mathcal{A}$.

4. Finally, on receiving $\mathcal{A}$'s guess σ': If $\sigma' = \sigma$, $\mathcal{B}$ outputs 1; else it outputs 0.

From the construction, we observe the following:

• If the oracle $\mathcal{B}$ can access is $U(\mathbb{Z}_q^{n+1})$, then $(b_0, b_1, \ldots, b_m)$ are independently uniform distributed over $\mathbb{Z}_q$, and hence $\mathbf{c}_{id^*}$ always follows the uniform distribution over $\mathbb{Z}_q^{m+1}$. So $\mathcal{A}$'s view in the environment simulated by $\mathcal{B}$ is similar to that in Game 5.

• If the oracle $\mathcal{B}$ can access is $A_{\mathbf{s},\chi}$, then, by Definition 1, the distribution of $\mathbf{c}_{id^*}$ in the experiment between $\mathcal{B}$ and $\mathcal{A}$ are identical with that in Game 4.

From the above observation, we have that the absolute difference of $\mathcal{A}$'s success probabilities in Game 4 and in Game 5 is just $\mathcal{B}$'s advantage, which is negligible by the assumption.

And by the above analysis, we prove that each two adjacent games are (at least) computationally indistinguishable and $\mathcal{A}$'s advantage in the final game is zero. Therefore, we conclude that $\mathcal{A}$'s advantage in Game 0 is negligible, which completes the proof.

Appendix D Proof of Lemma 1

Let $\hat{\mathbf{s}} = (\mathbf{s}_{id_1}^T || \ldots || \mathbf{s}_{id_N}^T)^T$, according to the decryption, we compute $\widehat{\mathbf{C}}_{id_i}\hat{\mathbf{s}}$ to obtain a vector consisting of N blocks, where the i-th block is $\mathbf{C}_{id_i}\mathbf{s}_{id_i}$ and each of other blocks is $\mathbf{X}_j\mathbf{s}_{id_i} + \mathbf{C}_{id_i}\mathbf{s}_{id_j}$ for $j \in [N]\backslash\{i\}$, where

$$\mathbf{C}_{id_i}\mathbf{s}_{id_i} = (\mathbf{S}_i\mathbf{A}_{id_i} + \mathbf{E} + \mu\mathbf{G}_{m+1})\mathbf{s}_{id_i} = \mathbf{E}\mathbf{s}_{id_i} + \mu\mathbf{G}_{m+1}\mathbf{s}_{id_i},$$

for $j \in [N]\backslash\{i\}$,

$$\begin{aligned}\mathbf{X}_j\mathbf{s}_{id_i} &= \sum_{a=1}^{(m+1)\ell_q}\sum_{b=1}^{n}\mathbf{G}_{m+1}^{-1}(\mathbf{U}_{a,b})\vec{\mathcal{S}}_i[a,b]\cdot\mathbf{s}_{id_i}\\ &= \sum_{a=1}^{(m+1)\ell_q}\sum_{b=1}^{n}(\mathbf{G}_{m+1}^{-1}(\mathbf{U}_{a,b})\mathbf{E}_{a,b}\mathbf{s}_{id_i}) + \mathbf{S}_i(\mathbf{u}_{id_j} - \mathbf{u}_{id_i})\\ &\leqslant n(m+1)^3\ell_q^2 B_\chi + \mathbf{S}_i(\mathbf{u}_{id_j} - \mathbf{u}_{id_i}),\end{aligned}$$

let $\mathbf{e}_j := \sum_{a=1}^{(m+1)\ell_q}\sum_{b=1}^{n}(\mathbf{G}_{m+1}^{-1}(\mathbf{U}_{a,b})\mathbf{E}_{a,b}\mathbf{s}_{id_i})$

$$\begin{aligned}\mathbf{X}_j\mathbf{s}_{id_i} + \mathbf{C}_{id_i}\mathbf{s}_{id_j} &= (\mathbf{e}_j + \mathbf{S}_i(\mathbf{u}_{id_j} - \mathbf{u}_{id_i})) + (\mathbf{S}_i(\mathbf{u}_{id_i} - \mathbf{u}_{id_j}) + \mathbf{E}\mathbf{s}_{id_j} + \mu\mathbf{G}_{m+1}\mathbf{s}_{id_j})\\ &= \mathbf{e}_j + \mathbf{E}\mathbf{s}_{id_j} + \mu\mathbf{G}_{m+1}\mathbf{s}_{id_j},\end{aligned}$$

here $\|\mathbf{E}\mathbf{s}_{id_j}\|_\infty \leqslant (m+1)B_\chi$, for $j \in [N]$, so the noise of $\widehat{\mathbf{C}}_{id_i}$ has magnitude $\mathcal{O}(m^4\ell_q B_\chi)$.

Appendix E A Fully Dynamic Multi-id IBFHE Scheme

After constructing our single-hop MIBFHE scheme, we can apply it to obtain a fully dynamic MIBFHE scheme by publishing the way of generating ciphertexts of specific users' secret keys in use of $i\mathcal{O}$ and then exploiting a multi-id bootstrapping technique similar to that in [13]. Note that for homomorphic evaluation, we need to construct an obfuscated program in setup phase, where SMIBFHE.Enc will be called. However, the subroutine is probabilistic, in which $i\mathcal{O}$ doesn't work. To solve the problem, we sample the randomness in advance to make it deterministic.

Let $\mathsf{F}' : \mathcal{K}' \times \{0,1\}^k \times [m+1] \to \mathcal{R}_{\mathsf{SampleS1}} \times \mathcal{R}_{\mathsf{SampleE1}} \times \mathcal{R}_{\mathsf{SampleS2}} \times \mathcal{R}_{\mathsf{SampleE2}}$ be another puncturable family of PRFs, $\mathcal{R}_{\mathsf{SampleS1}}$, $\mathcal{R}_{\mathsf{SampleE1}}$, $\mathcal{R}_{\mathsf{SampleS2}}$ and $\mathcal{R}_{\mathsf{SampleE2}}$ denote the randomness spaces of probabilistic algorithms SampleS1, SampleE1, SampleS2 and SampleE2, respectively.

• FMIBFHE.Setup$(\cdot)$: On input 1^λ, do the same as IBE.Setup(1^λ) except replacing $\mathsf{H} \leftarrow i\mathcal{O}(\lambda, \mathsf{RelateID})$ with $\overline{\mathsf{H}} \leftarrow i\mathcal{O}(\lambda, \mathsf{ProducePP})$, where ProducePP is shown in Figure E1. Output $\mathtt{mpk} := (\mathbf{A}, \overline{\mathsf{H}})$, $\mathtt{msk} = K$.

• FMIBFHE.KeyGen$(\cdot,\cdot)$: On input $\mathtt{msk}$ and $id \in \{0,1\}^k$, do the same as IBE.KeyGen$(\mathtt{msk}, id)$. Output $\mathbf{s}_{id} = (1, \mathbf{w}_{id}^T)^T$.

• FMIBFHE.Enc$(\cdot,\cdot,\cdot)$: On input $\mathtt{mpk}$, $id \in \{0,1\}^k$ and $\mu \in \{0,1\}$, do the same as IBE.Enc$(\mathtt{mpk}, id, \mu)$ except using $\overline{\mathsf{H}}(id)$ instead of $\mathsf{H}(id)$ to generate $(\mathbf{u}_{id}, \vec{\mathcal{S}}_{id})$, and using only $\mathbf{u}_{id}$ for encryption. Output $\mathbf{c}_{id} \in \mathbb{Z}_q^{m+1}$.

• FMIBFHE.Dec$(\cdot,\cdot)$: On input $(\mathbf{s}_{id_1}, \ldots, \mathbf{s}_{id_N})$ and $\hat{\mathbf{c}} \in \mathbb{Z}_q^{(m+1)N}$, denote $\hat{\mathbf{s}} = (\mathbf{s}_{id_1}^T || \ldots || \mathbf{s}_{id_N}^T)^T$ and compute $\delta := \langle \hat{\mathbf{c}}, \hat{\mathbf{s}} \rangle$. Output $\left\lfloor \frac{\delta}{q/2} \right\rceil$.

• FMIBFHE.Eval$(\cdot,\cdot,\cdot)$: On input $\mathtt{mpk}$, $f : \{0,1\}^t \to \{0,1\}$ and $(\mathbf{c}_1, \ldots, \mathbf{c}_t)$, assume, w.l.o.g., that these ciphertexts are under t different identities $id_1, \ldots, id_t$, respectively, and that f consists only of NAND gates with fan-in 2, which can be evaluated homomorphically gate by gate.

We can construct an augmented decryption function for each gate with two input ciphertexts $\hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2$ under two sequences of secret keys $\hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2$, respectively, as follows:

$$\mathsf{AD}_{\hat{\mathbf{c}}_1,\hat{\mathbf{c}}_2}(\hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2) = \mathsf{NAND}(\mathsf{FMIBFHE.Dec}(\hat{\mathbf{s}}_1, \hat{\mathbf{c}}_1), \mathsf{FMIBFHE.Dec}(\hat{\mathbf{s}}_2, \hat{\mathbf{c}}_2)).$$

Input: $id \in \{0,1\}^k$
Constants: $\mathbf{A}, K, K'$

1. $\mathbf{w}_{id} := \mathsf{F}(K, id) \in \{0,1\}^m$, $\mathbf{u}_{id} := \mathbf{A}\mathbf{w}_{id} \in \mathbb{Z}_q^n$, $\mathbf{s}_{id} := (1, \mathbf{w}_{id}^T)^T \in \{0,1\}^{m+1}$,
2. for $i \in [m+1]$,
   $(r_{1,i}, r_{2,i}, \{(r_{3,i,j}, r_{4,i,j})\}_{j\in[(m+1)n\ell_q]}) \leftarrow \mathsf{F}'(K', id, i)$,
   $\mathbf{S}_{1,i} := \mathsf{SampleS1}(\mathbb{Z}^{(m+1)\ell_q \times n}, \chi; r_{1,i})$,
   $\mathbf{E}_{1,i} := \mathsf{SampleE1}(\mathbb{Z}^{(m+1)\ell_q \times (m+1)}, \chi; r_{2,i})$,
   $\{\mathbf{S}_{2,i,j}\}_j := \mathsf{SampleS2}(\mathbb{Z}_q^{(m+1)\ell_q \times n}, U; \{r_{3,i,j}\}_j)$,
   $\{\mathbf{E}_{2,i,j}\}_j := \mathsf{SampleE2}(\mathbb{Z}^{(m+1)\ell_q \times (m+1)}, \chi; \{r_{4,i,j}\}_j)$,
   $\vec{\mathcal{S}}_{id}[i] \leftarrow \mathsf{SMIBFHE.Enc}((\mathbf{A}, \mathbf{u}_{id}), id, \mathbf{s}_{id}[i]; \mathbf{S}_{1,i}, \mathbf{E}_{1,i}, \{(\mathbf{S}_{2,i,j}, \mathbf{E}_{2,i,j})\}_{j\in[(m+1)n\ell_q]})$,
3. output $(\mathbf{u}_{id}, \vec{\mathcal{S}}_{id})$.

Figure E1 Program ProducePP

Table E1 The Comparison between Fully Dynamic MIBFHE Schemes

Scheme	Ciphertext Size	Encryption	Compactness	Security	Model
[14] initiated by [13] and [1]	$n^5\ell_q^5$	BP.KeyGen + BP.Enc + $n\ell_q$ times GPV.Enc	# of involving ciphertexts	Adaptive	RO
[14] initiated by [13] and [2]	$n^5\ell_q^5$	BP.KeyGen + BP.Enc + $n\ell_q$ times ABB.Enc	# of involving ciphertexts	Adaptive	Standard
Our scheme	$n\ell_q^2$	An obfuscated program + GPV.Enc	# of involving identities	Selective	Standard

From previous works [13, 19], we know that the function is in NC^1, and hence by the on-the-fly variant of Barrington's theorem from Corollary 2.3 in [13], we obtain the result ciphertext under $\hat{\mathbf{s}}$ for involving identities, with the help of $\overline{\mathsf{H}}(\cdot)$ on each involving identity and the ciphertext expansion in SMIBFHE.Eval.

Lemma 4 (Correctness)**.** For any polynomial N on λ and N different identities $id_1, \ldots, id_N$, any ciphertexts $\hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2$ of μ_1, μ_2 under two sequences of secret keys $\hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2 \subseteq \{\mathbf{s}_{id_1}, \ldots, \mathbf{s}_{id_N}\}$, respectively, then the following holds:

$$\mathsf{FMIBFHE.Dec}((\mathbf{s}_{id_1}, \ldots, \mathbf{s}_{id_N}), \mathsf{FMIBFHE.Eval}(\mathtt{mpk}, \mathsf{NAND}, (\hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2))) = \mathsf{NAND}(\mu_1, \mu_2),$$

and the evaluated ciphertext with $\mathcal{O}(LNm^5\ell_q^2 B_\chi)$, where $L = \mathrm{poly}(\lambda)$ is the length of the permutation branching program, $(\mathtt{mpk}, \mathtt{msk}) \leftarrow \mathsf{FMIBFHE.Setup}(1^\lambda)$, $\mathbf{s}_{id_i} \leftarrow \mathsf{FMIBFHE.KeyGen}(\mathtt{msk}, id_i)$ for $i \in [N]$.

Remark 1. It is not hard to prove the above lemma following that for Lemma 4 in [13]. The following conclusion can be similarly proved as for Theorem 1.

Theorem 1 (Security)**.** The fully-dynamic MIBFHE scheme FMIBFHE is IND-sID-CPA secure in the standard model if the scheme SMIBFHE is IND-sID-CPA secure and weakly circular secure, $i\mathcal{O}$ is an indistinguishability obfuscator, F and F' are two puncturable families of PRFs.

In Table E1, we present the comparison between fully dynamic MIBFHE schemes [14] initiated by [13] and [1], [2] respectively, and our scheme.

References

1 Gentry C, Peikert C, Vaikuntanathan V. Trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, 2008. 197-206.
2 Agrawal S, Boneh D, Boyen X. Efficient Lattice (H)IBE in the Standard Model. In: Proceedings of 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, 2010. 553-572.
3 Cash D, Hofheinz D, Kiltz E, et al. Bonsai Trees, or How to Delegate a Lattice Basis. In: Proceedings of 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, 2010. 523-552.
4 Singh K, Pandurangan C, Banerjee A K. Adaptively secure efficient lattice (H)IBE in standard model with short public parameters. In: Proceedings of International Conference on Security, Privacy, and Applied Cryptography Engineering. 2012. 153-172.
5 Zhang J, Chen Y, Zhang Z F. Programmable hash functions from lattices: Short signatures and IBEs with small key sizes. In: Proceddings of 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, 2016. 303-332.
6 Yamada S. Asymptotically compact adaptively secure lattice IBEs and verifiable random functions via generalized partitioning techniques. In: Proceedings of 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, 2017. 161-193.

7 Apon D, Fan X, Liu F H. Vector encoding over lattices and its applications. IACR Cryptology ePrint Arcgive, 2017: 455, 2017.

8 Gentry C, Sahai A, Waters B. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In: Proceedings of 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, 2013. 75-92.

9 Alperin-Sheriff J, Peikert C. Faster bootstrapping with polynomial error. In: Proceedings of the 34th Annual Cryptology Conference, Santa Barbara, CA, USA, 2014. 297-314.

10 Mukherjee P, Wichs D. Two round multiparty computation via multi-key FHE. In: Proceedings of 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, 2016. 735-763.

11 Regev O. On lattices, learning with errors, random linear codes, and cryptography. In: Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, 2005. 84-93.

12 Peikert C. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, Bethesda, MD, USA, 2009. 333-342.

13 Brakerski Z, Perlman R. Lattice-based fully dynamic multi-key FHE with short ciphertexts. In: Proceedings of 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, 2016. 190-213.

14 Canetti R, Raghuraman S, Richelson S, et al. Chosen-ciphertext secure fully homomorphic encryption. In: Proceedings of 20th IACR International Conference on Practice and Theory in Public-Key Cryptography, Amsterdam, The Netherlands, 2017. 213-240.

15 Garg S, Gentry C, Halevi S, et al. Candidate indistinguishability obfuscation and functional encryption for all circuits. In: Proceedings of 54th Annual IEEE Symposium on Foundations of Computer Science, Berkeley, CA, USA, 2013. 40-49.

16 Lin H J, Vaikuntanathan V. Indistinguishability obfuscation from DDH-like assumptions on constant-degree graded encodings. In: Proceedings of Symposium on Foundations of Computer Science, New Jersey, USA, 2016. 11-20.

17 Ananth P, Sahai A. Projective arithmetic functional encryption and indistinguishability obfuscation from degree-5 multilinear maps. In: Proceedings of 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, 2017. 152-181.

18 Sahai A, Waters B. How to use indistinguishability obfuscation: deniable encryption, and more. In: Proceedings of Symposium on Theory of Computing, New York, USA, 2014. 475-484.

19 Brakerski Z, Vaikuntanathan V. Efficient fully homomorphic encryption from (standard) LWE. In: Proceedings of 52nd Annual Symposium on Foundations of Computer Science, Palm Springs, CA, USA, 2011. 97-106.