

# Fully Dynamic Multi Target Homomorphic Attribute-Based Encryption

Ryo Hiromasa \*      Yutaka Kawai †

April 27, 2017

## Abstract

We propose multi target homomorphic attribute-based encryption (MT-HABE) with *fully dynamic* homomorphic evaluation: it can take as input *arbitrary* additional ciphertexts during homomorphic computation. In the previous MT-HABE of Brakerski et al. (TCC 2016-B), the output of homomorphic computation, which is related to a policy set, cannot be computed with a fresh ciphertext whose attribute does not satisfy any policy in the set. This is because the underlying multi-key fully homomorphic encryption (MKFHE) is single-hop: some keys are related to the output of homomorphic computation, which cannot be combined with ciphertexts encrypted under other keys. To implement fully dynamic homomorphic evaluations, we construct MT-HABE from the multi-hop MKFHE proposed by Peikert and Shiehian (TCC 2016-B).

---

\*Mitsubishi Electric. E-mail: [Hiromasa.Ryo@aj.MitsubishiElectric.co.jp](mailto:Hiromasa.Ryo@aj.MitsubishiElectric.co.jp)

†Mitsubishi Electric. E-mail: [Kawai.Yutaka@da.MitsubishiElectric.co.jp](mailto:Kawai.Yutaka@da.MitsubishiElectric.co.jp)

# 1 Introduction

Fully homomorphic encryption (FHE) allows us to evaluate any function over encrypted data by only using public information. Since the breakthrough work by Gentry [Gen09a, Gen09b], many different varieties of FHE have been proposed [DGHV10, BV11b, BV11a, BGV12, Bra12, LTV12, GSW13, CLT14]. FHE can be used, for example, to outsource computations to remote servers (e.g., cloud servers) without compromising privacy.

A cloud server may be used by multiple users, so it is required to set access permission among them. Attribute-based encryption (ABE) is a special type of public key encryption to accomplish this requirement. In key-policy ABE scheme, a (master) public key  $\text{mpk}$  is used to generate a ciphertext of a message  $\mu$ , which is labeled with a public attribute  $x \in \{0, 1\}^\ell$ . The secret key  $\text{sk}_f$  is associated to a policy  $f : \{0, 1\}^\ell \rightarrow \{0, 1\}$  and it can only decrypt ciphertexts that satisfy  $f(x) = 0$ . Previously, several ABE schemes under the learning with errors (LWE) assumption have been proposed [GVW13, BGG<sup>+</sup>14, BV16], and it was known that from [GSW13] and [GVW13] we can construct homomorphic ABE (HABE). The HABE scheme enables us to both set access permission and homomorphically evaluate on the ciphertexts, but the homomorphism is somewhat limited: the scheme can correctly evaluate only on the ciphertexts with the same attribute. In [CM16], Clear and McGoldrick proposed a way to compile the above HABE to an HABE with non-leveled homomorphism, but the resulting scheme still has the limitation over the attributes.

In [BCTW16], Brakerski et al. proposed target HABE (T-HABE) that enables cross-attribute homomorphic evaluations. A syntactical difference between T-HABE and HABE is in the homomorphic evaluation algorithm. In T-HABE, a homomorphic evaluation algorithm takes as input a set of policies  $F = \{f_i\}_i$ , an operation  $g$ , and some ciphertexts  $\{(ct_j, x_j)\}$ , where each ciphertext encrypts  $\mu_j$ . If for any  $x_j$  there exists  $f_i$  such that  $f_i(x_j) = 0$ , the algorithm outputs a ciphertext  $\text{ct}_F^{(g)}$  that can be decrypted by using all of the secret keys  $\{\text{sk}_{f_i}\}_i$ , and the result of the decryption is  $g(\{\mu_j\}_j)$  with high probability. The paper proposed two types of T-HABE, single target HABE (ST-HABE) and multi target HABE (MT-HABE). ST-HABE is an T-HABE that can homomorphically evaluate between the ciphertexts each of whose attributes satisfy a certain single policy, i.e., T-HABE in which  $F = \{f\}$  for a single policy  $f$ . In MT-HABE, a set of policies is related to the homomorphic computation, which can be processed between ciphertexts whose attribute satisfies some policy in the set. The MT-HABE of [BCTW16] is constructed from the ST-HABE and multi-key FHE (MKFHE) of [CM15, MW16].

The MT-HABE proposed in [BCTW16] is *not fully dynamic*: the output of ciphertexts, which depends on a certain policy set  $F$ , cannot be homomorphically evaluated with the fresh ciphertext whose attribute does not satisfy any policy in  $F$ . This forces the evaluator to know all the involved policies before the computation begins.

## 1.1 Our Results

We propose *fully dynamic* MT-HABE: it can take as input *arbitrary* additional ciphertexts during homomorphic computation. This enables us both dynamic cross-attribute homomorphic computations and setting access permissions.

In the previous MT-HABE of [BCTW16], the output of homomorphic computation is related to a policy set  $F$ , and it cannot be computed with a fresh ciphertext whose attribute does not satisfy any policy in  $F$ . This is because the underlying multi-key fully homomorphic encryption (MKFHE) is single-hop: some keys are related to the output of homomorphic computation, which cannot be combined with ciphertexts

	Key Size	Ciphertext size
[BCTW16]	$\tilde{O}(n^2 d_{\text{BCTW}}^2 \ell)$	$\tilde{O}(\ell n^4 d_{\text{BCTW}}^6) \rightarrow \tilde{O}(d^2 n^2 d_{\text{BCTW}}^4)$
Ours	$\tilde{O}(n^2 d_{\text{ours}}^2 \ell)$	$\tilde{O}(n^3(\ell + d_{\text{ours}})d_{\text{ours}}^6) \rightarrow \tilde{O}(n^3 d_{\text{ours}}^7)$

Table 1: Comparison of key and ciphertext size between the previous MT-HABE [BCTW16] and our fully dynamic MT-HABE. The parameter  $n$  is the LWE dimension,  $\ell$  is the maximal number of inputs of policies,  $d_{\text{BCTW}} = d_{\mathcal{F}} + d_{\mathcal{G}} \log d$ , and  $d_{\text{ours}} = d \log d + d_{\mathcal{G}} + d_{\mathcal{F}} \log \ell$ , where  $d$  represents the bound of the number of involving policies on homomorphic computations, and  $d_{\mathcal{F}}$  and  $d_{\mathcal{G}}$  denote the maximal depths of policies and operations, respectively. The left and right hand sides of the notation  $\rightarrow$  represent the size of a fresh and evaluated ciphertext.

encrypted under other keys. To implement dynamic homomorphic evaluation algorithms, we construct MT-HABE from the multi-hop MKFHE proposed by Peikert and Shiehian [PS16].

The security of the proposed MT-HABE is proven under the same assumption as [BCTW16]: the LWE assumption with sub-exponential modulus to noise ratio in the random oracle model. A comparison of key and ciphertext size between the MT-HABE of [BCTW16] and our scheme is shown in Table 1, which says that the size of the public key of our scheme is almost the same as [BCTW16] ignoring the logarithmic factor.

## 1.2 Our Techniques

We denote a vector in column form (so its transpose is in row form). Let  $n, q$  be LWE parameters,  $m = O(n \log q)$ ,  $N := n \lceil \log q \rceil$ ,  $M := (m + N + 1) \lceil \log q \rceil$ , and  $\mathbf{g}^T := (1, 2, 2^2, \dots, 2^{\lceil \log q \rceil})$ . In the following, we use the notation  $x \approx y$  to represent the noisy equation  $x = y + e$  for some noise term  $e$ . The starting point of the proposed scheme is the MKFHE scheme of [PS16].

**Multi-hop MKFHE of [PS16].** A ciphertext of the MKFHE is a triple of matrices  $(\mathbf{C}, \mathbf{F}, \mathbf{D})$  such that for a secret key vector  $\mathbf{t}$ ,

$$\mathbf{t}^T \mathbf{C} \approx \mu(\mathbf{t}^T \otimes \mathbf{g}^T), \quad \mathbf{F} = \hat{\mathbf{F}} + \mu(\mathbf{I}_n \otimes \mathbf{g}^T), \quad (\mathbf{I}_n \otimes \mathbf{t}^T) \mathbf{D} \approx (\mathbf{R} \otimes \mathbf{g}),$$

where  $\hat{\mathbf{F}} = \mathbf{A}\mathbf{R}$  for a random public matrix  $\mathbf{A}$  and a random binary matrix  $\mathbf{R}$ . To achieve dynamic homomorphism on the ciphertexts, the MKFHE has an algorithm to expand a ciphertext  $\mathbf{C}$  under  $\mathbf{t}$  into a ciphertext  $\mathbf{C}'$  under  $\mathbf{t}' = [\mathbf{t}, \mathbf{t}^*]$  for an additional key  $\mathbf{t}^*$ , where  $\mathbf{C}$  and  $\mathbf{C}'$  encrypts the same message. The expanded ciphertext  $\mathbf{C}'$  is generated by

$$\mathbf{C}' := \begin{bmatrix} \mathbf{C} & \mathbf{X} \\ & \mathbf{F} \end{bmatrix},$$

for a matrix  $\mathbf{X}$  that satisfies  $\mathbf{t}'^T \mathbf{X} + \mathbf{t}'^T \mathbf{A}\mathbf{R} \approx \mathbf{0}$ . Since it holds that  $\mathbf{t}'^T \mathbf{C}' \approx \mu(\mathbf{t}'^T \otimes \mathbf{g}^T)$ , which is the approximate eigenvector relation as in [GSW13], we can homomorphically evaluate on these expanded ciphertexts.

**ST-HABE of [BCTW16].** A public parameter contains random matrices  $\mathbf{A}, \mathbf{B}_0, \mathbf{B}_1, \dots, \mathbf{B}_\ell$  and a random vector  $\mathbf{v}$ . We define  $\vec{\mathbf{B}}_x := [\mathbf{B}_1, \dots, \mathbf{B}_\ell]$ , and  $x\vec{\mathbf{G}} := [x_1(\mathbf{I}_n \otimes \mathbf{g}^T), \dots, x_\ell(\mathbf{I}_n \otimes \mathbf{g}^T)]$  for an attribute

$x \in \{0, 1\}^\ell$ . A ciphertext of the ST-HABE consists of the following two matrices

$$\mathbf{C} \approx \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_0^T \\ \mathbf{v}^T \end{bmatrix} \cdot \mathbf{S} + \mu(\mathbf{I}_{m+N+1} \otimes \mathbf{g}^T), \quad \mathbf{C}_x \approx (\vec{\mathbf{B}}_x - x\vec{\mathbf{G}})^T \cdot \mathbf{S}$$

for some random matrix  $\mathbf{S}$ . Let  $f : \{0, 1\}^\ell \rightarrow \{0, 1\}$  be a policy and  $\mathbf{B}_f$  be a matrix generated upon  $f$ . The secret key for  $f$  is a vector  $\mathbf{r}_f$  such that  $\mathbf{r}_f^T \mathbf{A}^T + \mathbf{r}_f'^T (\mathbf{B}_0 + \mathbf{B}_f)^T + \mathbf{v}^T = \mathbf{0}$  for a random binary vector  $\mathbf{r}_f'$ , which is generated by the random oracle in the MT-HABE of [BCTW16]. There exists a matrix  $\mathbf{H}$  such that  $\mathbf{B}_f - f(x)(\mathbf{I}_n \otimes \mathbf{g}^T) = (\vec{\mathbf{B}}_x - x\vec{\mathbf{G}})\mathbf{H}$ . In homomorphic evaluations, the ST-HABE generates the functioned ciphertext for the policy  $f$  by computing  $\hat{\mathbf{C}}_f := \mathbf{C} + [\mathbf{0}_{M \times m}, \mathbf{C}_x^T \mathbf{H}, \mathbf{0}_M]^T$ . If  $f(x) = 0$  holds, then the functioned ciphertext satisfies the approximate eigenvector relation of [GSW13] with  $\mathbf{t}_f^T = [\mathbf{r}_f^T, \mathbf{r}_f'^T, 1]$ .

**Our Scheme.** We construct fully dynamic MT-HABE by making the multi-hop MKFHE [PS16] attribute-based. In the scheme of [PS16], the matrix  $\mathbf{F}$  contains a message  $\mu$ , so  $\hat{\mathbf{F}}$  must be indistinguishable from uniform to ensure the security. The matrix  $\hat{\mathbf{F}}$  is set to be  $\mathbf{A}\mathbf{R}$ , which is statistically indistinguishable from uniform by the leftover hash lemma (LHL). To make [PS16] attribute-based, it is required to use the matrix  $[\mathbf{A}, \mathbf{B}, \mathbf{v}]^T$  instead of just  $\mathbf{A}$ , but the number of rows of  $[\mathbf{A}, \mathbf{B}, \mathbf{v}]^T$  is insufficient for saying  $[\mathbf{A}, \mathbf{B}, \mathbf{v}]^T \mathbf{R}$  indistinguishable from uniform by LHL. To deal with this problem, we set  $\hat{\mathbf{F}} = [\mathbf{A}, \mathbf{B}, \mathbf{v}]^T \mathbf{R} + \mathbf{E} \approx [\mathbf{A}, \mathbf{B}, \mathbf{v}]^T \mathbf{R}$ , and say that the matrix is computationally indistinguishable from uniform by the LWE assumption.

In the proposed MT-HABE, the functioned ciphertext is computed in a similar way to [BCTW16], and it consists of the following three matrices such that for a secret key  $\mathbf{t}$ ,

$$\mathbf{t}^T \mathbf{C} \approx \mu(\mathbf{t}^T \otimes \mathbf{g}^T), \quad \mathbf{F} \approx \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}^T \\ \mathbf{v}^T \end{bmatrix} \cdot \mathbf{R} + \mu(\mathbf{I}_{m+N+1} \otimes \mathbf{g}^T), \quad (\mathbf{I}_N \otimes \mathbf{t}^T) \mathbf{D} \approx (\mathbf{R} \otimes \mathbf{g}),$$

where  $m = O(n \log^2 q)$  for the security reason. To dynamically evaluate on this ciphertext, we need to implement the ciphertext expansion algorithm, which transforms the ciphertext  $\mathbf{C}$  under the key  $\mathbf{t}$  to the ciphertext  $\mathbf{C}'$  under  $[\mathbf{t}, \mathbf{t}_f]$  for an additional policy  $f$ . The algorithm must compute a matrix  $\mathbf{X}$  such that  $\mathbf{t}^T \mathbf{X} + \mathbf{t}_f^T \mathbf{F} \approx \mu(\mathbf{t}_f^T \otimes \mathbf{g}^T)$ , which is in other words  $\mathbf{t}^T \mathbf{X} + \mathbf{t}_f^T [\mathbf{A}, \mathbf{B}, \mathbf{v}]^T \cdot \mathbf{R} \approx \mathbf{0}$ . However, the term  $\mathbf{r}_f^T \mathbf{A}$ , which is from expanding  $\mathbf{t}_f^T [\mathbf{A}, \mathbf{B}, \mathbf{v}]^T$ , cannot be known because  $\mathbf{r}_f$  is a part of the secret key. To overcome this problem, our algorithm instead computes  $\mathbf{X}$  such that  $\mathbf{t}^T \mathbf{X} \approx \mathbf{r}_f'^T (\mathbf{B}_0 + \mathbf{B}_f - \mathbf{B})^T \cdot \mathbf{R}$ , where  $\mathbf{r}_f'^T$  is obtained from the random oracle, the matrices  $\mathbf{B}_0$  and  $\mathbf{B}$  are the public matrices, and  $\mathbf{B}_f$  can publicly be generated from  $f$ . Then, it holds that

$$\begin{aligned} \mathbf{t}^T \mathbf{X} + \mathbf{t}_f^T \mathbf{F} &\approx \mathbf{r}_f'^T \cdot (\mathbf{B}_0 + \mathbf{B}_f - \mathbf{B})^T \cdot \mathbf{R} + [\mathbf{r}_f^T, \mathbf{r}_f'^T, 1] \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}^T \\ \mathbf{v}^T \end{bmatrix} \cdot \mathbf{R} + \mu(\mathbf{t}_f^T \otimes \mathbf{g}^T) \\ &= [\mathbf{r}_f^T, \mathbf{r}_f'^T, 1] \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_0^T + \mathbf{B}_f^T \\ \mathbf{v}^T \end{bmatrix} \cdot \mathbf{R} + \mu(\mathbf{t}_f^T \otimes \mathbf{g}^T) \\ &= \mu(\mathbf{t}_f^T \otimes \mathbf{g}^T). \end{aligned}$$

### 1.3 Organization

In Section 2, we introduce mathematical preliminaries used in this paper. In Section 3, we show the construction of the proposed fully dynamic MT-HABE. In Section 4, we prove the correctness and security of our construction.

## 2 Preliminaries

**Notations.** We denote the set of natural numbers by  $\mathbb{N}$ , and the set of integers by  $\mathbb{Z}$ . For any positive integer  $d > 0$ , we represent  $\{1, 2, \dots, d\}$  by  $[d]$ . Let  $S$  be a set and  $\mathcal{P}$  be a probability distribution over  $S$ . Then, we denote by  $a \leftarrow S$  that  $a \in S$  is chosen uniformly at random from  $S$ , and by  $b \leftarrow \mathcal{P}$  that  $b \in S$  is sampled from  $\mathcal{P}$ . The notation  $\text{negl}(\lambda)$  represents the set of negligible functions for  $\lambda \in \mathbb{N}$ .

Vectors are in column form and written by bold lower-case letters (e.g.,  $\mathbf{x}$ ). The  $i$ -th element of the vector  $\mathbf{x}$  is represented by  $x_i$ . We denote the  $\ell_\infty$  norm (max norm) of the vector  $\mathbf{x}$  by  $\|\mathbf{x}\|_\infty$ . The inner-product of two vectors is written by  $\langle \mathbf{x}, \mathbf{y} \rangle$ . We denote matrices as the bold capital letters (e.g.,  $\mathbf{X}$ ) and the  $i$ -th column vector of the matrix  $\mathbf{X}$  is represented by  $\mathbf{X}[i]$ . For matrix  $\mathbf{X} \in \mathbb{R}^{m \times n}$ , the  $\ell_\infty$  norm of  $\mathbf{X}$  is defined as  $\|\mathbf{X}\|_\infty := \max_{i \in [n]} \{\|\mathbf{X}[i]\|_\infty\}$ . The notation  $\mathbf{X}^T \in \mathbb{R}^{n \times m}$  represents the transpose of  $\mathbf{X}$ . For two matrices  $\mathbf{A} \in \mathbb{R}^{m \times n_1}$  and  $\mathbf{B} \in \mathbb{R}^{m \times n_2}$ ,  $[\mathbf{A}, \mathbf{B}] \in \mathbb{R}^{m \times (n_1 + n_2)}$  is the matrix generated by concatenating  $\mathbf{A}$  and  $\mathbf{B}$ . Let  $\mathbf{I}_n$  be the  $n \times n$  identity matrix, and  $\mathbf{0}_{n \times m}$  be the  $n \times m$  matrix all of whose entries are 0. For any  $i \in [n]$ ,  $\mathbf{u}_i \in \{0, 1\}^n$  represents the  $i$ -th standard basis vector of dimension  $n$ .

**Tensor Products.** The tensor product of an  $m_1 \times n_1$  matrix  $\mathbf{A}$  and  $m_2 \times n_2$  matrix  $\mathbf{B}$  over a commutative ring  $\mathcal{R}$  is the  $m_1 m_2 \times n_1 n_2$  matrix consisting of  $m_2 \times n_2$  blocks whose  $(i, j)$ -th block is  $a_{i,j} \mathbf{B}$ , where  $a_{i,j}$  is the  $(i, j)$ -th element of  $\mathbf{A}$ .

For any scalar  $r \in \mathcal{R}$ , we have

$$r(\mathbf{A} \otimes \mathbf{B}) = (r\mathbf{A}) \otimes \mathbf{B} = \mathbf{A} \otimes (r\mathbf{B}).$$

We heavily use the mixed product property of tensor products, which say

$$(\mathbf{A} \otimes \mathbf{B}) \cdot (\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD})$$

for any matrices  $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$  with compatible dimensions. In particular, it holds that

$$\begin{aligned} \mathbf{A} \otimes \mathbf{B} &= (\mathbf{A} \otimes \mathbf{I}_{\text{height}(\mathbf{B})}) \cdot (\mathbf{I}_{\text{width}(\mathbf{A})} \otimes \mathbf{B}) \\ &= (\mathbf{I}_{\text{height}(\mathbf{A})} \otimes \mathbf{B}) \cdot (\mathbf{A} \otimes \mathbf{I}_{\text{width}(\mathbf{B})}). \end{aligned}$$

**Noisy Equations.** In this paper, we consider the noisy equations, and we use the notation  $\approx$  to say that the two sides of the equation are approximately equal within some additive error. For example,

$$x \approx y \quad (\text{error: } B)$$

represents  $x = y + e$  for some  $e \in [-B, B]$ .

## 2.1 Target Homomorphic Attribute-Based Encryption

In [BCTW16], Brakerski et al. first introduced the notion of target homomorphic attribute based encryption (T-HABE), which is an homomorphic encryption whose homomorphic operations depend on policies. We define the syntax of T-HABE and then define its correctness and security.

**Definition 2.1** (Target Homomorphic Attribute Based Encryption (T-HABE)). *A target homomorphic attribute based encryption scheme consists of five algorithms  $\text{THABE} = \text{THABE}.\{\text{Setup}, \text{Enc}, \text{Keygen}, \text{Dec}, \text{Eval}\}$  with the following syntax.*

- $\text{THABE.Setup}(1^\lambda)$  : takes as input a security parameter  $\lambda$  (additionally, the algorithm can take parameters that specifies classes of policies or admissible operations), and outputs a public parameter  $\text{pp}$  and master secret key  $\text{msk}$ .
- $\text{THABE.Enc}_{\text{pp}}(\mu, x)$  : takes as input a public parameter  $\text{pp}$ , plaintext  $\mu$ , and attribute  $x$ , and outputs a tuple of a ciphertext and attribute  $(\text{ct}, x)$ .
- $\text{THABE.Keygen}_{\text{msk}}(f)$  : takes as input a master secret key  $\text{msk}$  and policy  $f$ , and outputs a secret key  $\text{sk}_f$ .
- $\text{THABE.Eval}_{\text{pp}}(F, \text{ct}^{(1)}, \dots, \text{ct}^{(k)}, g)$  : takes as input a public parameter  $\text{pp}$ , set of policies  $F$ , ciphertexts  $\text{ct}^{(1)}, \dots, \text{ct}^{(k)}$ , and a function  $g$ , and outputs a ciphertext  $\text{ct}^g$ .
- $\text{THABE.Dec}_{\text{sk}_F}(\text{ct}^g)$  : takes as input a secret key  $\text{sk}_F$  ( $\text{sk}_F = \{\text{sk}_f : f \in F\}$ ) and ciphertext  $\text{ct}^g$ , and outputs a plaintext  $\mu \in \{0, 1\}$ .

The correctness of T-HABE guarantees that the ciphertext is correctly decrypted to the intended value with high probability when given all the keys for the policies involving in the homomorphic computation.

**Definition 2.2** (Correctness of T-HABE). *Let  $\{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$  be a class of policies, and  $\{\mathcal{G}_\lambda\}_{\lambda \in \mathbb{N}}$  be class of operations. The T-HABE scheme  $\text{THABE} = \text{THABE}.\{\text{Setup}, \text{Enc}, \text{Keygen}, \text{Eval}, \text{Dec}\}$  is correct if the following holds.*

*Let  $(\text{pp}, \text{msk}) \leftarrow \text{THABE.Setup}(1^\lambda)$ . Consider a set of  $\text{poly}(\lambda)$  policy  $F \subseteq \mathcal{F}_\lambda$ , set of the corresponding secret keys  $\text{sk}_F := \{\text{sk}_f : f \in F\}$ , a sequence of  $k \geq 1$  messages and attributes  $\{(\mu^{(i)} \in \{0, 1\}, x^{(i)} \in \{0, 1\}^*)\}_{i \in [k]}$  such that  $\forall x^{(i)}, \exists f \in F, f(x^{(i)}) = 0$ , their ciphertexts  $\{\text{ct}^{(i)} \leftarrow \text{THABE.Enc}_{\text{pp}}(\mu^{(i)}, x^{(i)})\}_{i \in [k]}$ . Then, computing  $\text{ct}^g := \text{THABE.Eval}_{\text{pp}}(F, \text{ct}^{(1)}, \dots, \text{ct}^{(k)}, g)$  for some  $g \in \mathcal{G}$ , it holds that*

$$\Pr[\text{THABE.Dec}_{\text{sk}_F}(\text{ct}^g) \neq g(\mu^{(1)}, \dots, \mu^{(k)})] = \text{negl}(\lambda),$$

where the probability is take over the randomness in the experiment.

The security is defined in the same way as standard (key-policy) ABE.

**Definition 2.3** (Security of T-HABE). *Let  $\text{THABE}$  be a T-HABE scheme described in the above, and consider the following game between the challenger and adversary.*

1. *The adversary sends an attribute  $x^*$  to the challenger.*
2. *The challenger generates  $(\text{msk}, \text{pp}) \leftarrow \text{THABE.Setup}(1^\lambda)$  and sends  $\text{pp}$  to the adversary.*

3. The adversary makes arbitrary many key generation queries by sending  $f_i$  (represented as circuits) to the challenger. Upon receiving such functions, the challenger creates a key  $\text{sk}_{f_i} \leftarrow \text{THABE.Keygen}_{\text{msk}}(f_i)$  and sends  $\text{sk}_{f_i}$  if  $f_i(x^*) = 1$ , and sends  $\perp$  otherwise.
4. The adversary sends a pair of messages  $\mu_0, \mu_1$  to the challenger. The challenger chooses  $b \leftarrow \{0, 1\}$  uniformly at random, and computes  $\text{ct}^* \leftarrow \text{THABE.Enc}_{\text{pp}}(\mu_b, x^*)$ . It sends  $\text{ct}^*$  to the challenger.
5. The adversary makes arbitrary many key generation queries as in Step 3.
6. The adversary outputs  $b' \in \{0, 1\}$ .

The above game is called the selective security game, and the advantage of the adversary in this game is defined by  $\text{Adv}_{\mathcal{A}}^{\text{SS-THABE}}(\lambda) := |\Pr[b' = b] - 1/2|$ , where  $b$  and  $b'$  are generated in the game. The scheme THABE is selectively secure if for any PPT adversary  $\mathcal{A}$ , it holds that  $\text{Adv}_{\mathcal{A}}^{\text{SS-THABE}}(\lambda) = \text{negl}(\lambda)$ .

As well as the previous attribute-based encryption from lattices, we allow decryption when  $f(x) = 0$ , and all of the queries must satisfy  $f_i(x^*) = 1$ .

## 2.2 Learning with Errors (LWE)

The *Learning with errors (LWE)* assumption was first introduced by Regev [Reg05]. The decision version of the LWE problem is called Decisional LWE (DLWE) and defined as follows.

**Definition 2.4 (DLWE).** For a security parameter  $\lambda$ , let  $n := n(\lambda)$  be a integer lattice,  $q := q(\lambda) \geq 2$  be a integer modulus, and  $\chi := \chi(\lambda)$  be a error distribution over  $\mathbb{Z}$ .  $\text{DLWE}_{n,q,\chi}$  is the problem to distinguish the following two distributions: in the first distribution, a pair  $(\mathbf{a}_i, b_i)$  is sampled from the uniform distribution over  $\mathbb{Z}_q^n \times \mathbb{Z}_q$ . In the second distribution, a pair  $(\mathbf{a}_i, b_i)$  is sampled by sampling  $\mathbf{a}_i \leftarrow \mathbb{Z}_q^n$  and  $\mathbf{s} \leftarrow \mathbb{Z}_q^n$  from the uniform distribution,  $e_i \leftarrow \chi$  from the noise distribution, and set  $b_i := \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i \pmod q$ .  $\text{DLWE}_{n,q,\chi}$  assumption states that  $\text{DLWE}_{n,q,\chi}$  is intractable for any PPT adversary.

By letting  $\chi$  be a discrete Gaussian distribution over  $\mathbb{Z}$  with parameter  $r = \alpha q \geq 2\sqrt{n}$  (represented by  $D_{\mathbb{Z},r}$ ) for some  $0 < \alpha < 1$ , there exists a quantum reduction [Reg05] from  $\text{DLWE}_{n,q,\chi=D_{\mathbb{Z},r}}$  to approximating a short vector over  $n$  dimensional lattices within factor of  $\tilde{O}(n/\alpha)$ <sup>1</sup>. Additionally, it is known that there exists the classical reductions [Pei09, BLP<sup>+</sup>13] for other parameters.

## 2.3 Gadget Matrix and Bit Decomposition

Let  $\mathbf{g}^T := (1, 2, \dots, 2^{\lceil \log q \rceil})$  be a vector consisting of the powers of 2. The operation  $\mathbf{g}^{-1} : \mathbb{Z}_q \rightarrow \{0, 1\}^{1 \times \lceil \log q \rceil}$  takes as input  $x \in \mathbb{Z}_q$ , and outputs  $\mathbf{y}$  such that  $\langle \mathbf{y}, \mathbf{g} \rangle = x \in \mathbb{Z}_q$ . For example,  $\mathbf{g}^{-1}$  is the operation to decompose  $x$  into its binary representation. Symmetrically,  $\mathbf{g}^{-T} : \mathbb{Z}_q \rightarrow \{0, 1\}^{\lceil \log q \rceil}$  transforms an element in  $\mathbb{Z}_q$  into the column vector of its binary representation. More generally, the operation  $(\mathbf{I}_n \otimes \mathbf{g}^{-T})(\cdot)$  generates  $n \cdot \lceil \log q \rceil$  dimensional vector with coefficients of  $\{0, 1\}$  by applying  $\mathbf{g}^{-T}$  to every element of the vector in  $\mathbb{Z}_q^n$ . Then the following holds

$$(\mathbf{I}_n \otimes \mathbf{g}^T) \cdot (\mathbf{I}_n \otimes \mathbf{g}^{-T})(\mathbf{x}) = \mathbf{x}.$$

It is clear that this operation can be generalized to matrices.

<sup>1</sup> Approximating a short vector over  $n$  dimensional lattices within factor of  $\gamma$  takes  $2^{\tilde{O}(n/\log \gamma)}$  computations [Sch87].

## 2.4 Lattice Trapdoors and Discrete Gaussian Distributions

Consider a matrix  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ . For all  $\mathbf{V} \in \mathbb{Z}_q^{n \times m'}$  and for any probability distribution  $P$  over  $\mathbb{Z}^m$ , let  $\mathbf{A}_P^{-1}(\mathbf{V})$  be the random variable whose distribution is  $P$  conditioned on  $\mathbf{A} \cdot \mathbf{A}_P^{-1}(\mathbf{V}) = \mathbf{V}$ . A  $P$ -trapdoor for  $\mathbf{A}$  is an algorithm that can efficiently sample from a distribution within  $2^{-n}$  statistical distance of  $\mathbf{A}_P^{-1}(\mathbf{V})$  for any  $\mathbf{V}$ . We denote the  $P$ -trapdoor by  $\mathbf{A}_P^{-1}$ , and  $\mathbf{A}_P^{-1} = \mathbf{A}_\tau^{-1}$  in the case where  $P$  is a Gaussian distribution with parameter  $\tau$ .

In the following, we introduce the procedures to generate an almost uniform  $\mathbf{A}$  with a trapdoor for sampling from the Gaussian distribution.

**Corollary 2.1** (Generating Trapdoors [GPV08, MP12, BLP<sup>+</sup>13]). *There exists an efficient algorithm  $\text{TrapGen}(1^n, q, m)$  that outputs  $(\mathbf{A}, \mathbf{A}_{\tau_0}^{-1})$ , where  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$  for any  $m \geq m_0$  for  $m_0 = O(n \log q)$ ,  $\mathbf{A}$  is statistically close to uniform over  $\mathbb{Z}_q^{n \times m}$  within  $2^{-n}$  distance, and  $\tau_0 = O(\sqrt{n \log q \log n})$ . Given  $\mathbf{A}_{\tau_0}^{-1}$ , one can obtain  $\mathbf{A}_\tau^{-1}$  for any  $\tau \geq \tau_0$ .*

**Corollary 2.2** (Gaussian-Binary Sampler [LW15]). *Let  $n, m, q$  be such that  $m \geq n \lceil \log q \rceil$ . With all but  $O(2^{-n})$  probability over the choice of  $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ , for all  $\mathbf{R} \in \mathbb{Z}^{m \times N}$  with  $N = n \lceil \log q \rceil$ , one can obtain  $[\mathbf{A}, \mathbf{A}\mathbf{R} + (\mathbf{I}_n \otimes \mathbf{g}^T)]_P^{-1}$  with  $P = D_{\mathbb{Z}^m, \tau} \times \{0, 1\}^N$  for  $\tau = O(N\sqrt{mn} \cdot \|\mathbf{R}\|_\infty)$ . Furthermore, for all  $\mathbf{v}$ , it holds that the marginal distribution of the last  $N$  coordinates of  $[\mathbf{A}, \mathbf{A}\mathbf{R} + (\mathbf{I}_n \otimes \mathbf{g}^T)]_P^{-1}(\mathbf{v})$  is statistically close to uniform over  $\{0, 1\}^N$  within  $2^{-n}$  distance.*

## 2.5 Homomorphic Operations

Here we define the procedure used for homomorphic evaluations in our scheme.

**Definition 2.5.** *Let  $n, q, \ell \in \mathbb{N}$  and  $N := n \lceil \log q \rceil$ . Consider  $\mathbf{B}_1, \dots, \mathbf{B}_\ell \in \mathbb{Z}_q^{n \times N}$ , and denote  $\vec{\mathbf{B}} := [\mathbf{B}_1, \dots, \mathbf{B}_\ell]$ . Let  $f$  be a Boolean circuit of depth  $d$  that computes a function  $\{0, 1\}^\ell \rightarrow \{0, 1\}$  and consists only of NAND gates. We define  $\mathbf{B}_f := \text{Eval}(\vec{\mathbf{B}}, f)$  recursively: associate  $\mathbf{B}_1, \dots, \mathbf{B}_\ell$  with the  $\ell$  input wires of  $f$ . For every wire  $w \in f$ , let  $u, v$  be its predecessors and define*

$$\mathbf{B}_w := (\mathbf{I}_n \otimes \mathbf{g}^T) - \mathbf{B}_u \cdot (\mathbf{I}_n \otimes \mathbf{g}^{-T})(\mathbf{B}_v).$$

Finally,  $\mathbf{B}_f$  is the matrix associated with the output wire of  $f$ .

The following fact represents the properties of the above homomorphic evaluation algorithm.

**Fact 2.1.** *Consider  $\mathbf{B}_1, \dots, \mathbf{B}_\ell \in \mathbb{Z}_q^{n \times N}$  ( $N = n \lceil \log q \rceil$ ). Letting  $\vec{\mathbf{B}} := [\mathbf{B}_1, \dots, \mathbf{B}_\ell]$ , and  $x\vec{\mathbf{G}} := [x_1(\mathbf{I}_n \otimes \mathbf{g}^T), \dots, x_\ell(\mathbf{I}_n \otimes \mathbf{g}^T)]$ , there exists a polynomial time algorithm  $\text{EvRelation}$  such that if  $\mathbf{H} := \mathbf{H}_{f, x, \vec{\mathbf{B}}} := \text{EvRelation}(f, x, \vec{\mathbf{B}})$ , then  $\|\mathbf{H}\|_\infty \leq (N + 1)^d$  and*

$$(\mathbf{B}_f - f(x)(\mathbf{I}_n \otimes \mathbf{g}^T))^T = \mathbf{H}^T \cdot [\vec{\mathbf{B}} - x\vec{\mathbf{G}}]^T,$$

where  $\mathbf{B}_f = \text{Eval}(f, \vec{\mathbf{B}})$ .

*In particular, if  $\mathbf{B}_i := \mathbf{A}\mathbf{R}_i + x_i(\mathbf{I}_n \otimes \mathbf{g}^T)$ , that is,  $\vec{\mathbf{B}} = \mathbf{A}\vec{\mathbf{R}} + x\vec{\mathbf{G}}$  for  $\vec{\mathbf{R}} := [\mathbf{R}_1, \dots, \mathbf{R}_\ell]$ , then  $\mathbf{B}_f = \mathbf{A}\mathbf{R}_f + f(x)(\mathbf{I}_n \otimes \mathbf{g}^T)$  for  $\mathbf{R}_f = \vec{\mathbf{R}} \cdot \mathbf{H}_{f, x, \vec{\mathbf{B}}}$ .*

We can see that this fact holds by verifying that for the NAND operation in Definition 2.5,

$$\text{EvRelation}(\text{NAND}, (x_u, x_v), [\mathbf{B}_u, \mathbf{B}_v]) = \begin{bmatrix} -(\mathbf{I}_n \otimes \mathbf{g}^{-T})(\mathbf{B}_v) \\ -x_u \mathbf{I}_{N \times N} \end{bmatrix}.$$



### 3 Fully Dynamic MT-HABE

In this section, we construct fully dynamic MT-HABE from the multi-hop MKFHE scheme of [PS16]. The proposed MT-HABE can take as input arbitrary additional ciphertexts during homomorphic computations. We show the construction in Section 3.1 except for homomorphic evaluation algorithms, which are described in Section 3.2.

#### 3.1 Construction

Let  $\mathcal{F} \subseteq \{0, 1\}^\ell \rightarrow \{0, 1\}$  be a class of policies computed by depth- $d_{\mathcal{F}}$  circuits only from NAND gates, and  $\mathcal{G} \subseteq \{0, 1\}^* \rightarrow \{0, 1\}$  be a class of operations computed by depth- $d_{\mathcal{G}}$  circuits only from NAND gates. Let  $\text{PRF}.\{\text{Gen}, \text{Eval}\}$  be a pseudorandom function, and  $d$  be the designed bound of the number of involving policies on homomorphic computations.

- $\text{dMTHABE.Setup}(1^\lambda, 1^\ell, 1^{d_{\mathcal{F}}}, 1^{d_{\mathcal{G}}}, 1^d)$  : choose DLWE parameters  $n, q, \chi$  as described in Section 4.1. Let  $B$  be a bound of samples from error distribution  $\chi$ . Let  $m = O(n \log^2 q)$ ,  $N := n \lceil \log q \rceil$ , and  $M := (m + N + 1) \lceil \log q \rceil$ . Generate  $(\mathbf{A}, \mathbf{A}_{\tau_0}^{-1}) \leftarrow \text{TrapGen}(1^n, q, m)$ , where  $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$  and  $\tau_0 = O(\sqrt{n \log q \log n})$  from Corollary 2.1. Sample random matrices  $\mathbf{B}, \mathbf{B}_0, \mathbf{B}_1, \dots, \mathbf{B}_\ell \leftarrow \mathbb{Z}_q^{n \times N}$ , and let  $\vec{\mathbf{B}}_x := [\mathbf{B}_1, \dots, \mathbf{B}_\ell]$ . Sample a random vector  $\mathbf{v} \leftarrow \mathbb{Z}_q^n$ . Choose a PRF seed  $\sigma \leftarrow \text{PRF.Gen}(1^\lambda)$ . Let  $H : \mathbb{Z}_q^{n \times m} \times \mathcal{F} \rightarrow \{0, 1\}^N$  be a hash function implemented by the random oracle. Output  $\text{pp} := (\mathbf{A}, \mathbf{B}, \mathbf{B}_0, \vec{\mathbf{B}}_x, \mathbf{v}, H)$  and  $\text{msk} := (\mathbf{A}_{\tau_0}^{-1}, \sigma)$ .
- $\text{dMTHABE.Enc}_{\text{pp}}(\mu \in \{0, 1\}, x \in \{0, 1\}^\ell)$  : sample a random matrix  $\mathbf{S} \leftarrow \mathbb{Z}_q^{n \times M}$ , error matrix  $\mathbf{E}_A \leftarrow \chi^{m \times M}$ , and error vector  $\mathbf{e}_v \leftarrow \chi^M$ . For every  $i \in \{0, 1, \dots, \ell\}$  and  $j \in [M]$ , sample  $\mathbf{R}_{i,j} \leftarrow \{0, 1\}^{m \times N}$ , define  $\mathbf{E}_i[j] := \mathbf{R}_{i,j}^T \mathbf{E}_A[j]$ , and compute

$$\mathbf{C} := \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_0^T \\ \mathbf{v}^T \end{bmatrix} \cdot \mathbf{S} + \begin{bmatrix} \mathbf{E}_A \\ \mathbf{E}_0 \\ \mathbf{e}_v^T \end{bmatrix} + \mu(\mathbf{I}_{m+N+1} \otimes \mathbf{g}^T) \in \mathbb{Z}_q^{(m+N+1) \times M}$$

$$\mathbf{C}_x := (\vec{\mathbf{B}}_x - x\vec{\mathbf{G}})^T \cdot \mathbf{S} + \begin{bmatrix} \mathbf{E}_1 \\ \vdots \\ \mathbf{E}_\ell \end{bmatrix} \in \mathbb{Z}_q^{\ell N \times M}.$$

Choose a random matrix  $\mathbf{R} \leftarrow \mathbb{Z}_q^{n \times M}$  and sample a noise matrix  $\mathbf{E}_A^{(F)} \leftarrow \chi^{m \times M}$ . For every  $j \in [M]$ , choose  $\mathbf{R}_j^{(F)} \leftarrow \{0, 1\}^{m \times N}$  and define  $\mathbf{E}^{(F)}[j] := (\mathbf{R}_j^{(F)})^T \mathbf{E}_A^{(F)}[j]$ . Sample  $\mathbf{e}_v^{(F)} \leftarrow \chi^M$ , and compute

$$\mathbf{F} := \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}^T \\ \mathbf{v}^T \end{bmatrix} \cdot \mathbf{R} + \begin{bmatrix} \mathbf{E}_A^{(F)} \\ \mathbf{E}^{(F)} \\ (\mathbf{e}_v^{(F)})^T \end{bmatrix} + \mu(\mathbf{I}_{m+N+1} \otimes \mathbf{g}^T) \in \mathbb{Z}_q^{(m+N+1) \times M}$$

$$\approx \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}^T \\ \mathbf{v}^T \end{bmatrix} \cdot \mathbf{R} + \mu(\mathbf{I}_{m+N+1} \otimes \mathbf{g}^T) \quad (\text{error: } mB).$$

For every  $i \in \{0, 1, \dots, \ell\}$ ,  $j \in [M]$ , and  $k \in [N]$ , sample  $\mathbf{E}_A^{(k)} \leftarrow \chi^{m \times M}$  and  $\mathbf{e}_v^{(k)} \leftarrow \chi^M$ , compute  $\mathbf{E}_i^{(k)}[j] := \mathbf{R}_{i,j}^T \mathbf{E}_A^{(k)}[j]$ , and set  $\mathbf{E}^{(k)} := [(\mathbf{E}_A^{(k)})^T, (\mathbf{E}_0^{(k)})^T, \mathbf{e}_v^{(k)T}]$ . Sample  $\mathbf{S}^{(1)}, \dots, \mathbf{S}^{(N)} \leftarrow \chi^{n \times M}$ , and

compute

$$\mathbf{D} := \left( \mathbf{I}_N \otimes \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_0^T \\ \mathbf{v}^T \end{bmatrix} \right) \cdot \begin{bmatrix} \mathbf{S}^{(1)} \\ \vdots \\ \mathbf{S}^{(N)} \end{bmatrix} + \begin{bmatrix} \mathbf{E}^{(1)} \\ \vdots \\ \mathbf{E}^{(N)} \end{bmatrix} + \mathbf{R} \otimes \mathbf{g} \otimes \mathbf{u}_{m+N+1} \in \mathbb{Z}_q^{(m+N+1)N \times M},$$

$$\mathbf{D}_x^{(k)} := (\vec{\mathbf{B}}_x - x\vec{\mathbf{G}})^T \cdot \mathbf{S}^{(k)} + \begin{bmatrix} \mathbf{E}_1^{(k)} \\ \vdots \\ \mathbf{E}_\ell^{(k)} \end{bmatrix} \in \mathbb{Z}_q^{\ell N \times M}.$$

Output  $\text{ct} := (x, \mathbf{C}, \mathbf{C}_x, \mathbf{F}, \mathbf{D}, \{\mathbf{D}_x^{(k)}\}_k)$ .

- **dMTHABE.Keygen<sub>mSk</sub>**( $f \in \mathcal{F}$ ) : compute  $\mathbf{B}_f := \text{Eval}(f, \vec{\mathbf{B}}_x)$ . Generate  $\mathbf{r}'_f = H(\mathbf{A}, f) \in \{0, 1\}^N$  by using the random oracle. Sample  $\mathbf{r}_f \leftarrow \mathbf{A}_\tau^{-1}(-(\mathbf{B}_0 + \mathbf{B}_f)\mathbf{r}'_f - \mathbf{v}; \rho)$  with randomness  $\rho \leftarrow \text{PRF.Eval}(\sigma, f)$ , where  $\tau = O(\sqrt{mn} \cdot N^2 \ell(N+1)^{d_{\mathcal{F}}}) \geq \tau_0$ . Then, it holds that

$$[\mathbf{r}_f^T, \mathbf{r}'_f{}^T, 1] \begin{bmatrix} \mathbf{A}^T \\ (\mathbf{B}_0 + \mathbf{B}_f)^T \\ \mathbf{v}^T \end{bmatrix} = \mathbf{0}_{1 \times n}.$$

Output  $\text{sk}_f := \mathbf{r}_f$ .

- **dMTHABE.ApplyF<sub>pp</sub>**( $\text{ct}, f \in \mathcal{F}$ ) : when given  $\text{ct}$  and  $f$ , first compute  $\mathbf{H} := \text{EvRelation}(f, x, \vec{\mathbf{B}}_x)$ . Set  $\mathbf{C}_f := \mathbf{H}^T \mathbf{C}_x$ , and compute

$$\hat{\mathbf{C}}_f := \mathbf{C} + \begin{bmatrix} \mathbf{0}_{m \times M} \\ \mathbf{C}_f \\ \mathbf{0}_{1 \times M} \end{bmatrix}.$$

Then it holds that for a secret key  $\mathbf{t}_f^T := [\mathbf{r}_f^T, \mathbf{r}'_f{}^T, 1]$  with related to  $f$ ,

$$\mathbf{t}_f^T \cdot \hat{\mathbf{C}}_f \approx \mu(\mathbf{t}_f^T \otimes \mathbf{g}^T) \quad (\text{error: } \|\mathbf{t}_f\|_\infty \cdot ((N+1)^{d_{\mathcal{F}}} \cdot \ell N + 1) \cdot mB).$$

For every  $k \in [N]$ , let  $\mathbf{D}_f^{(k)} := \mathbf{H}^T \mathbf{D}_x^{(k)}$ , and

$$\mathbf{D}_f := [\mathbf{0}_{M \times m}, (\mathbf{D}_f^{(1)})^T, \mathbf{0}_M, \dots, \mathbf{0}_{M \times m}, (\mathbf{D}_f^{(N)})^T, \mathbf{0}_M]^T.$$

Compute  $\hat{\mathbf{D}}_f := \mathbf{D} + \mathbf{D}_f$ . It holds that for the secret key  $\mathbf{t}_f$ ,

$$(\mathbf{I}_N \otimes \mathbf{t}_f^T) \cdot \hat{\mathbf{D}}_f \approx \mathbf{R} \otimes \mathbf{g} \in \mathbb{Z}_q^{N \times M} \quad (\text{error: } \|\mathbf{t}_f\|_\infty \cdot ((N+1)^{d_{\mathcal{F}}} \cdot \ell N + 1) \cdot mB).$$

Output the functioned ciphertext  $\text{ct}^{(f)} := (\hat{\mathbf{C}}_f, \mathbf{F}, \hat{\mathbf{D}}_f)$ .

- **dMTHABE.Eval**( $\text{ct}^{(1)}, \dots, \text{ct}^{(k)}, F := \{f_1, \dots, f_d\} \subseteq \mathcal{F}, g \in \mathcal{G}$ ) : for every  $i \in [k]$ , there exists  $f_j$  such that  $f_j(x_i) = 0$ . For all  $i \in [k]$ , compute a functioned ciphertext **dMTHABE.ApplyF<sub>pp</sub>**( $\text{ct}^{(i)}, f_j$ ), and homomorphically evaluate  $g$  between the functioned ciphertexts. Output  $\text{ct}^{(F)}$ .

- **dMTHABE.Dec**<sub>sk<sub>f<sub>1</sub></sub>, ..., sk<sub>f<sub>d</sub></sub></sub>(ct<sup>(F)</sup>) : given secret keys sk<sub>f<sub>1</sub></sub>, ..., sk<sub>f<sub>d</sub></sub> for every policy in  $F = \{f_1, \dots, f_d\}$ , and an evaluated ciphertext ct<sup>(F)</sup> = ( $\hat{\mathbf{C}}_F, \mathbf{F}, \hat{\mathbf{D}}_F$ ) for  $F$ , first, for all  $j \in [d]$ , obtain  $\mathbf{r}'_{f_j} := H(\mathbf{A}, f_j)$  by using the random oracle. Construct the concatenated key  $\mathbf{t}_F^T := [\mathbf{r}'_{f_1}, \mathbf{r}'_{f_1}, 1, \dots, \mathbf{r}'_{f_d}, \mathbf{r}'_{f_d}, 1]$ , and compute a vector  $\mathbf{c} := \mathbf{t}_F^T \hat{\mathbf{C}}_F$ . Let  $\mathbf{u}^T := (0, \dots, 0, \lfloor q/2 \rfloor) \in \mathbb{Z}^{1 \times d(m+N+1)}$ . Compute  $\tilde{\mu} := \mathbf{c}^T \cdot (\mathbf{I}_{d(m+N+1)} \otimes \mathbf{g}^{-T})(\mathbf{u})$ , and output 0 if  $|\tilde{\mu}| < q/4$ , and 1 otherwise.

Correctness and security of this scheme are discussed in Appendix 4.

### 3.2 The Algorithm Eval

We here describe the algorithms used in homomorphic evaluation of Eval.

Suppose that we obtain a functioned ciphertext ct<sup>(f)</sup> := ( $\hat{\mathbf{C}}_f, \mathbf{F}, \hat{\mathbf{D}}_f$ ) by applying **dMTHABE.ApplyF** for a policy  $f \in \mathcal{F}$  to a fresh ciphertext ct := ( $x, \mathbf{C}, \mathbf{C}_x, \mathbf{F}, \mathbf{D}, \{\mathbf{D}_x^{(k)}\}_k$ ). Then the functioned ciphertext ct<sup>(f)</sup> satisfies the following three noisy equations with a secret key  $\mathbf{t}_f \in \mathbb{Z}_q^{m+N+1}$  for  $f$  and small random matrix  $\mathbf{R} \in \mathbb{Z}_q^{n \times M}$ . For ease of notation, let  $B_{\mathbf{C}}, B_{\mathbf{F}}, B_{\mathbf{D}}$  be bounds of errors included in  $\hat{\mathbf{C}}_f, \mathbf{F}$ , and  $\hat{\mathbf{D}}_f$ , respectively.

$$\mathbf{t}_f^T \hat{\mathbf{C}}_f \approx \mu(\mathbf{t}_f^T \otimes \mathbf{g}^T) \quad (\text{error: } B_{\mathbf{C}}) \quad (1)$$

$$\mathbf{F} \approx \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}^T \\ \mathbf{v}^T \end{bmatrix} \mathbf{R} + \mu(\mathbf{I}_{m+N+1} \otimes \mathbf{g}^T) \quad (\text{error: } B_{\mathbf{F}}) \quad (2)$$

$$(\mathbf{I}_N \otimes \mathbf{t}_f^T) \cdot \hat{\mathbf{D}}_f \approx \mathbf{R} \otimes \mathbf{g} \quad (\text{error: } B_{\mathbf{D}}). \quad (3)$$

**Ciphertext Expansion.** We describe a way to expand ciphertexts so that they can be decrypted by the concatenation of all the keys related to the target policies. This expansion method is very similar to that of [PS16]. Given ciphertext ( $\hat{\mathbf{C}}, \mathbf{F}, \hat{\mathbf{D}}$ ) that satisfies the three relations (1), (2), and (3) for secret key  $\mathbf{t} \in \mathbb{Z}_q^{n'}$  ( $n' = k(m+N+1)$  for some positive integer  $k$ ) and random matrix  $\mathbf{R} \in \mathbb{Z}_q^{n \times M}$ , generate ( $\tilde{\mathbf{C}}, \tilde{\mathbf{F}}, \tilde{\mathbf{D}}$ ) that satisfies the relations (1), (2), and (3) for the concatenated secret key  $\tilde{\mathbf{t}} := [\mathbf{t}, \mathbf{t}_f]$  consisted from  $\mathbf{t}$  and  $\mathbf{t}_f := [\mathbf{r}'_f, \mathbf{r}'_f, 1]^T \in \mathbb{Z}_q^{m+N+1}$ , and random matrix  $\tilde{\mathbf{R}}$ :

- $\mathbf{F}$  and  $\mathbf{R}$  are not changed. That is,  $\tilde{\mathbf{F}} := \mathbf{F}$  and  $\tilde{\mathbf{R}} := \mathbf{R}$ . This preserves the relation (2).
- $\tilde{\mathbf{D}}$  is computed as

$$\tilde{\mathbf{D}} := \left( \mathbf{I}_N \otimes \begin{bmatrix} \mathbf{I}_{n'} \\ \mathbf{0}_{(m+N+1) \times n'} \end{bmatrix} \right) \cdot \hat{\mathbf{D}}.$$

Then, since the following holds, the relation (3) is preserved.

$$\begin{aligned} (\mathbf{I}_N \otimes \tilde{\mathbf{t}}^T) \cdot \tilde{\mathbf{D}} &= (\mathbf{I}_N \otimes \mathbf{t}^T) \cdot \hat{\mathbf{D}} \\ &\approx \mathbf{R} \otimes \mathbf{g} \quad (\text{error: } B_{\mathbf{D}}). \end{aligned}$$

- We define

$$\tilde{\mathbf{C}} := \begin{bmatrix} \hat{\mathbf{C}} & \mathbf{X} \\ & \mathbf{F} \end{bmatrix},$$

where  $\mathbf{X}$  is a matrix computed by the following procedure. Let  $\mathbf{B}, \mathbf{B}_0 \in \mathbb{Z}_q^{n \times N}$  be matrices included in the public parameter, generate  $\mathbf{r}'_f = H(\mathbf{A}, f) \in \{0, 1\}^N$ , and compute  $\mathbf{B}_f := \text{Eval}(\tilde{\mathbf{B}}, f)$ . Define

$$\begin{aligned} \mathbf{s} &:= (\mathbf{I}_n \otimes \mathbf{g}^{-T})((\mathbf{B}_0 + \mathbf{B}_f - \mathbf{B})\mathbf{r}'_f) \in \{0, 1\}^N \\ \mathbf{X} &:= (\mathbf{s}^T \otimes \mathbf{I}_{n'}) \cdot \hat{\mathbf{D}}. \end{aligned}$$

Then, by construction of  $\mathbf{X}$ , it holds that

$$\begin{aligned} \mathbf{t}^T \mathbf{X} &= \mathbf{t}^T \cdot (\mathbf{s}^T \otimes \mathbf{I}_{n'}) \cdot \hat{\mathbf{D}} \\ &= (\mathbf{s}^T \otimes 1) \cdot (\mathbf{I}_N \otimes \mathbf{t}^T) \cdot \hat{\mathbf{D}} \\ &\approx \mathbf{s}^T \cdot \mathbf{R} \otimes \mathbf{g} \quad (\text{error: } N \cdot B_{\mathbf{D}}) \\ &= \mathbf{s}^T \cdot (\mathbf{I}_n \otimes \mathbf{g}) \cdot (\mathbf{R} \otimes 1) \\ &= \mathbf{r}'_f{}^T \cdot (\mathbf{B}_0 + \mathbf{B}_f - \mathbf{B})^T \cdot \mathbf{R}. \end{aligned}$$

From

$$\mathbf{t}^T \mathbf{X} + \mathbf{t}_f \mathbf{F}$$

$$\begin{aligned} &\approx \mathbf{r}'_f{}^T \cdot (\mathbf{B}_0 + \mathbf{B}_f - \mathbf{B})^T \cdot \mathbf{R} + [\mathbf{r}'_f{}^T, \mathbf{r}'_f{}^T, 1] \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}^T \\ \mathbf{v}^T \end{bmatrix} \cdot \mathbf{R} + \mu(\mathbf{t}_f^T \otimes \mathbf{g}^T) \quad (\text{error: } N \cdot B_{\mathbf{D}} + \|\mathbf{t}_f\|_{\infty} \cdot (m + N + 1) \cdot B_{\mathbf{F}}) \\ &= [\mathbf{r}'_f{}^T, \mathbf{r}'_f{}^T, 1] \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_0^T + \mathbf{B}_f^T \\ \mathbf{v}^T \end{bmatrix} \cdot \mathbf{R} + \mu(\mathbf{t}_f^T \otimes \mathbf{g}^T) \\ &= \mu(\mathbf{t}_f^T \otimes \mathbf{g}^T) \end{aligned}$$

we have

$$\tilde{\mathbf{t}}^T \tilde{\mathbf{C}} \approx \mu(\tilde{\mathbf{t}}^T \otimes \mathbf{g}^T) \quad (\text{error: } B_{\mathbf{C}} + N \cdot B_{\mathbf{D}} + \|\mathbf{t}_f\|_{\infty} \cdot (m + N + 1) \cdot B_{\mathbf{F}}),$$

and so the relation (1) is preserved for  $\tilde{\mathbf{C}}$ .

**Homomorphic Operations.** We here describe a way to evaluate homomorphic addition and multiplication. Consider two ciphertexts  $(\mathbf{C}_1, \mathbf{F}_1, \mathbf{D}_1)$  and  $(\mathbf{C}_2, \mathbf{F}_2, \mathbf{D}_2)$  that encrypt  $\mu_1, \mu_2 \in \{0, 1\}$  under the secret key  $\mathbf{t} \in \mathbb{Z}_q^{n'}$ . The two ciphertexts satisfy the relations (1), (2), and (3) for two random matrices  $\mathbf{R}_1, \mathbf{R}_2$ , respectively.

- Homomorphic addition: to homomorphically add the ciphertexts, we just add the corresponding matrices:

$$(\mathbf{C}_{\text{add}}, \mathbf{F}_{\text{add}}, \mathbf{D}_{\text{add}}) := (\mathbf{C}_1 + \mathbf{C}_2, \mathbf{F}_1 + \mathbf{F}_2, \mathbf{D}_1 + \mathbf{D}_2).$$

It is immediate that the relations (1), (2), and (3) are preserved for message  $\mu_{\text{add}} := \mu_1 + \mu_2$  and random matrix  $\mathbf{R}_{\text{add}} := \mathbf{R}_1 + \mathbf{R}_2$ .

- Homomorphic multiplication: to homomorphically multiply the ciphertexts, we compute the ciphertext consisting of the matrices computed as follows:

$$\begin{aligned} \mathbf{C}_{\text{mult}} &:= \mathbf{C}_1 \cdot (\mathbf{I}_{n'} \otimes \mathbf{g}^{-T})(\mathbf{C}_2) \\ \mathbf{F}_{\text{mult}} &:= \mathbf{F}_1 \cdot (\mathbf{I}_{m+N+1} \otimes \mathbf{g}^{-T})(\mathbf{F}_2) \\ \mathbf{D}_{\text{mult}} &:= \mathbf{D}_1 \cdot (\mathbf{I}_{m+N+1} \otimes \mathbf{g}^{-T})(\mathbf{F}_2) + (\mathbf{I}_N \otimes \mathbf{C}_1) \cdot (\mathbf{I}_{n'N} \otimes \mathbf{g}^{-T})(\mathbf{D}_2). \end{aligned}$$

We now show that the ciphertext output by the homomorphic multiplication procedure satisfies the relations (1), (2), and (3). Since  $\mathbf{C}_{mult}$  is the ciphertext output by the homomorphic multiplication of GSW FHE [GSW13], it is easy to see that the relation (1) is preserved. If we let  $B_{C_i}$  be an upper bound of the noise included in  $\mathbf{C}_i (i = 1, 2)$ , then we have

$$\begin{aligned} \mathbf{t}^T \mathbf{C}_{mult} &\approx \mu_1(\mathbf{t}^T \otimes \mathbf{g}^T) \cdot (\mathbf{I}_{n'} \otimes \mathbf{g}^{-T})(\mathbf{C}_2) \quad (\text{error: } n' \lceil \log q \rceil B_{C_1}) \\ &= \mu_1 \mathbf{t}^T \mathbf{C}_2 \\ &\approx \mu_1 \mu_2 (\mathbf{t}^T \otimes \mathbf{g}^T) \quad (\text{error: } \mu_1 B_{C_2}). \end{aligned}$$

Let  $\mathbf{R}_{mult} := \mathbf{R}_1 \cdot (\mathbf{I}_{m+N+1} \otimes \mathbf{g}^{-T})(\mathbf{F}_2) + \mu_1 \mathbf{R}_2$  and  $\mu_{mult} := \mu_1 \mu_2$ . Then the relation (2) is also preserved for  $\mathbf{F}_{mult}$ :

$$\begin{aligned} \mathbf{F}_{mult} &= \mathbf{F}_1 \cdot (\mathbf{I}_{m+N+1} \otimes \mathbf{g}^{-T})(\mathbf{F}_2) \\ &\approx \left( \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}^T \\ \mathbf{v}^T \end{bmatrix} \cdot \mathbf{R}_1 + \mu_1 (\mathbf{I}_{m+N+1} \otimes \mathbf{g}^T) \right) (\mathbf{I}_{m+N+1} \otimes \mathbf{g}^{-T})(\mathbf{F}_2) \quad (\text{error: } M \cdot B_{F_1}) \\ &\approx \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}^T \\ \mathbf{v}^T \end{bmatrix} \cdot (\mathbf{R}_1 \cdot (\mathbf{I}_{m+N+1} \otimes \mathbf{g}^{-T})(\mathbf{F}_2) + \mu_1 \mathbf{R}_2) + \mu_1 \mu_2 (\mathbf{I}_{m+N+1} \otimes \mathbf{g}^T) \quad (\text{error: } \mu_1 B_{F_2}) \\ &= \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}^T \\ \mathbf{v}^T \end{bmatrix} \cdot \mathbf{R}_{mult} + \mu_{mult} (\mathbf{I}_{m+N+1} \otimes \mathbf{g}^T). \end{aligned}$$

We check that the relation (3) is also preserved. First, we can see that

$$\begin{aligned} &(\mathbf{I}_N \otimes \mathbf{t}) \cdot \mathbf{D}_1 \cdot (\mathbf{I}_{m+N+1} \otimes \mathbf{g}^{-T})(\mathbf{F}_2) \\ &\approx (\mathbf{R}_1 \otimes \mathbf{g}) \cdot (\mathbf{I}_{m+N+1} \otimes \mathbf{g}^{-T})(\mathbf{F}_2) \quad (\text{error: } M \cdot B_{D_1}) \\ &= (\mathbf{R}_1 \cdot (\mathbf{I}_{m+N+1} \otimes \mathbf{g}^{-T})(\mathbf{F}_2)) \otimes \mathbf{g}. \end{aligned}$$

In addition, the following holds:

$$\begin{aligned} &(\mathbf{I}_N \otimes \mathbf{t})(\mathbf{I}_N \otimes \mathbf{C}_1) \cdot (\mathbf{I}_{n'} \otimes \mathbf{g}^{-T})(\mathbf{D}_2) \\ &= (\mathbf{I}_N \otimes \mathbf{tC}_1) \cdot (\mathbf{I}_{n'} \otimes \mathbf{g}^{-T})(\mathbf{D}_2) \\ &\approx \mu_1 (\mathbf{I}_N \otimes \mathbf{t}^T \otimes \mathbf{g}^T) \cdot (\mathbf{I}_{n'} \otimes \mathbf{g}^{-T})(\mathbf{D}_2) \quad (\text{error: } n' \lceil \log q \rceil B_{C_1}) \\ &= \mu_1 (\mathbf{I}_N \otimes \mathbf{t}^T) \cdot \mathbf{D}_2 \\ &\approx (\mu_1 \mathbf{R}) \otimes \mathbf{g} \quad (\text{error: } \mu_1 B_{D_2}). \end{aligned}$$

Hence, by

$$(\mathbf{I}_N \otimes \mathbf{t}^T) \mathbf{D}_{mult} \approx \mathbf{R}_{mult} \otimes \mathbf{g} \quad (\text{error: } M \cdot B_{D_1} + n' \lceil \log q \rceil B_{C_1} + \mu_1 B_{D_2}),$$

$\mathbf{D}_{mult}$  satisfies the relation (3).

## 4 Correctness and Security

In this section, we discuss about correctness and security of the proposed MT-HABE described in Section 3. In Section 4.1, we consider parameter settings of the proposed scheme for the correctness and security, and the proofs of them are described in Section 4.2.

## 4.1 Parameter Settings

The DLWE parameters  $n, q, \chi$  are chosen according to the conditions decided by the correctness and security.

It is required to set  $n \geq \lambda$  and  $q \leq 2^n$ . We also set  $\ell, d = \text{poly}(\lambda)$ . We estimate the worst-case noise growth when homomorphically evaluating a depth- $d_{\mathcal{G}}$  circuit consisting only of the NAND gate under  $d$  different policies of depth at most  $d_{\mathcal{F}}$ . We define the max error  $B_{\max}$  of the ciphertext  $(\mathbf{C}, \mathbf{F}, \mathbf{D})$  output by the algorithm ApplyF or Eval:

$$B_{\max} := \max(B_{\mathbf{C}}, B_{\mathbf{F}}, B_{\mathbf{D}}).$$

From the last section, the ciphertext generated by homomorphically evaluating a NAND gate has noise at most

$$\begin{aligned} & M \cdot B_{\mathbf{D}_1} + d(m + N + 1)[\log q]B_{\mathbf{C}_1} + \mu_1 B_{\mathbf{D}_2} \\ & \leq \{M \cdot (d + 1) + 1\} \cdot B_{\max} \\ & = \text{poly}(d, n, \lceil \log q \rceil) \cdot B_{\max}. \end{aligned}$$

for some polynomial  $\text{poly}(\cdot)$ . The ciphertext generated by the ciphertext expansion algorithm described in the last section also has noise at most

$$\begin{aligned} & B_{\mathbf{C}} + N \cdot B_{\mathbf{D}} + \|\mathbf{t}_f\|_{\infty} \cdot (m + N + 1) \cdot B_{\mathbf{F}} \\ & \leq (1 + N + \|\mathbf{t}_f\|_{\infty} \cdot (m + N + 1)) \cdot B_{\max} \\ & = \text{poly}'(n, \lceil \log q \rceil) \cdot B_{\max}. \end{aligned}$$

for some polynomial  $\text{poly}'(\cdot)$ .

Since the max error  $B_{\max}$  of fresh functioned ciphertexts is at most  $\|\mathbf{t}_f\|_{\infty} \cdot ((N + 1)^{d_{\mathcal{F}}} \cdot \ell N + 1)mB$ , the noise of the evaluated ciphertexts obtained by homomorphic evaluation of a depth- $d_{\mathcal{G}}$  circuit under different  $d$  policies is at most

$$\begin{aligned} & \text{poly}(d, n, \lceil \log q \rceil)^d \cdot \text{poly}'(n, \lceil \log q \rceil)^{d_{\mathcal{G}}} \cdot \|\mathbf{t}_f\|_{\infty} \cdot ((N + 1)^{d_{\mathcal{F}}} \cdot \ell N + 1)mB \\ & \leq \text{poly}(d, n, \lceil \log q \rceil)^d \cdot \text{poly}'(n, \lceil \log q \rceil)^{d_{\mathcal{G}}} \cdot O(\sqrt{mn} \cdot N^2 \cdot \ell(N + 1)^{d_{\mathcal{F}}}) \cdot \sqrt{m} \cdot ((N + 1)^{d_{\mathcal{F}}} \cdot \ell N + 1)mB. \end{aligned}$$

For the correctness and security, we select the parameters so that the above quantity by a factor of eight is less than  $2^{n^\epsilon}$  for some  $0 < \epsilon < 1$ . To hold this, we set  $n = \tilde{O}(d \cdot \log d + d_{\mathcal{G}} + d_{\mathcal{F}} \cdot \log \ell)^{1/\epsilon}$  and choose  $q$  and  $\chi$  so that they satisfy  $q/B \geq 2^{n^\epsilon}$ , where  $B$  is the upper bound of the noise distribution  $\chi$ . Selecting such parameters leads the reduction from the DLWE $_{n,q,\chi}$  problem to approximate a short vector on the  $n$  dimensional lattice by a factor of  $\tilde{O}(n \cdot 2^{n^\epsilon})$ .

## 4.2 Proofs

Correctness of our dMTHABE scheme are stated on the following theorem.

**Theorem 4.1** (Correctness). *The scheme dMTHABE with parameters  $\ell, d_{\mathcal{F}}, d_{\mathcal{G}}, d$  is correct for policy class  $\mathcal{F}_{\ell, d_{\mathcal{F}}}$  and homomorphism class  $\mathcal{G}_{d_{\mathcal{G}}}$ .*

*Proof.* Since correctness of the Eval algorithm is described in Section 3.2, and the parameters for correctly decrypting the evaluated ciphertexts are discussed in Section 4.1, we omit the proof of this theorem.  $\square$

Security of our dMTHABE can be proven similar to [BCTW16].

**Theorem 4.2 (Security).** *The scheme dMTHABE scheme is selectively secure for function classes  $\mathcal{F}, \mathcal{G}$  in the random oracle model if the DLWE $_{n,q,\chi}$  assumption holds.*

*Proof.* In a similar way to [BCTW16], we prove this theorem by considering about the indistinguishability of a column vector in the challenge ciphertext  $\mathbf{C}, \mathbf{C}_{x^*}, \mathbf{F}, \mathbf{D}, \{\mathbf{D}_{x^*}^{(k)}\}_{k \in [N]}$ , where we let  $x^*$  be the challenge attribute. That is, we consider the game in which the adversary is given the following vectors

$$\mathbf{c} := \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_0^T \\ \mathbf{v}^T \end{bmatrix} \cdot \mathbf{s} + \begin{bmatrix} \mathbf{e}_A \\ \mathbf{e}_0 \\ e_v \end{bmatrix}, \mathbf{c}_{x^*} := (\vec{\mathbf{B}}_{x^*} - x^* \vec{\mathbf{G}})^T \cdot \mathbf{s} + \begin{bmatrix} \mathbf{e}_1 \\ \vdots \\ \mathbf{e}_\ell \end{bmatrix}, \mathbf{f} := \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}^T \\ \mathbf{v}^T \end{bmatrix} \cdot \mathbf{r} + \begin{bmatrix} \mathbf{e}_A^{(F)} \\ \mathbf{e}_A^{(F)} \\ e_v^{(F)} \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{d}^{(1)} \\ \vdots \\ \mathbf{d}^{(N)} \end{bmatrix} := \mathbf{d} := \left( \mathbf{I}_N \otimes \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_0^T \\ \mathbf{v}^T \end{bmatrix} \right) \cdot \begin{bmatrix} \mathbf{s}^{(1)} \\ \vdots \\ \mathbf{s}^{(N)} \end{bmatrix} + \begin{bmatrix} \mathbf{e}^{(1)} \\ \vdots \\ \mathbf{e}^{(N)} \end{bmatrix}, \mathbf{d}_{x^*}^{(k)} := (\vec{\mathbf{B}}_{x^*} - x^* \vec{\mathbf{G}})^T \cdot \mathbf{s}^{(k)} + \begin{bmatrix} \mathbf{e}_1^{(k)} \\ \vdots \\ \mathbf{e}_\ell^{(k)} \end{bmatrix} \quad (\forall k \in [N]).$$

or the uniformly random vectors, and distinguishes them. We call this game *column game*, and define the advantage of the adversary in this game as  $\text{Adv}_{\mathcal{A}}^{\text{column}}(\lambda)$ . Without loss of generality, we can prove the security in the column game instead of proving the selective security game defined in Definition 2.3.

We now consider the following sequence of games. Let  $\text{Adv}_{\mathcal{A}}^{\text{Game}_i}(\lambda)$  be the advantage of the adversary  $\mathcal{A}$  in Game $_i$ .

- Game $_0$ : This game is the same as the column game, so it holds that

$$\text{Adv}_{\mathcal{A}}^{\text{column}}(\lambda) = \text{Adv}_{\mathcal{A}}^{\text{Game}_0}(\lambda).$$

- Game $_1$ : This game is the same as Game $_0$  except that the challenger aborts if the adversary sends the random oracle query  $(\mathbf{D}, f)$  such that  $\mathbf{D} = \mathbf{A}$  and  $f(x^*) = 1$  before the challenger outputs the challenge attribute  $x^*$ .

Since the probability that the adversary sends such query is  $\text{negl}(\lambda)$ , we have

$$|\text{Adv}_{\mathcal{A}}^{\text{Game}_1}(\lambda) - \text{Adv}_{\mathcal{A}}^{\text{Game}_0}(\lambda)| = \text{negl}(\lambda).$$

- Game $_2$ : This game is the same as Game $_1$  except that for every Keygen query the challenger uniformly chooses the randomness and use it for  $\mathbf{A}_{\tau_0}^{-1}$  instead of generating the randomness for  $\mathbf{A}_{\tau_0}^{-1}$  by using PRF. To answer the oracle query consistently, the challenger stores the Keygen query and its secret key to the table. By the property of the PRF, this game is indistinguishable from Game $_1$ :

$$|\text{Adv}_{\mathcal{A}}^{\text{Game}_2}(\lambda) - \text{Adv}_{\mathcal{A}}^{\text{Game}_1}(\lambda)| = \text{negl}(\lambda).$$

- Game $_3$ : This game is the same as Game $_2$  except for the generation of the public parameters  $\mathbf{B}, \mathbf{B}_0, \mathbf{B}_1, \dots, \mathbf{B}_\ell$ . Here, there exist matrices  $\mathbf{R}_0, \mathbf{R}_1, \dots, \mathbf{R}_\ell$  such that they are distributed uniformly over  $\{0, 1\}^{m \times N}$  and satisfies  $\mathbf{e}_i = \mathbf{R}_i^T \mathbf{e}_A$  and  $\mathbf{e}_i^{(k)} = \mathbf{R}_i^T \mathbf{e}_A^{(k)}$ . There exists a matrix  $\mathbf{R}^{(F)}$  such that it is distributed uniformly over  $\{0, 1\}^{m \times N}$  and satisfies  $\mathbf{e}^{(F)} = (\mathbf{R}^{(F)})^T \mathbf{e}_A^{(F)}$ . In this game, the public matrices  $\mathbf{B}, \mathbf{B}_0, \mathbf{B}_1, \dots, \mathbf{B}_\ell$  are computed as  $\mathbf{B} := \mathbf{A} \mathbf{R}^{(F)}, \mathbf{B}_0 := \mathbf{A} \mathbf{R}_0, \mathbf{B}_i := \mathbf{A} \mathbf{R}_i + x_i^* (\mathbf{I}_n \otimes \mathbf{g}^T)$  ( $\forall i \in [\ell]$ ) instead of choosing them uniformly at random. By the leftover hash lemma, every distribution of  $\mathbf{B}, \mathbf{B}_0, \mathbf{B}_1, \dots, \mathbf{B}_\ell$  is indistinguishable from uniform over  $\mathbb{Z}_q^{n \times N}$ . Hence we have

$$|\text{Adv}_{\mathcal{A}}^{\text{Game}_3}(\lambda) - \text{Adv}_{\mathcal{A}}^{\text{Game}_2}(\lambda)| = \text{negl}(\lambda).$$

- **Game<sub>4</sub>**: This game is the same as **Game<sub>3</sub>** except that the return  $\text{sk}_f$  for the key generation query  $(\mathbf{A}, f)$  is generated without using the trapdoor  $\mathbf{A}_\tau^{-1}$ .

Without loss of generality, we can assume that the tuple  $(\mathbf{A}, f)$  is queried to the **Keygen** oracle before querying to the random oracle. By the definition of selective security, the policy  $f$  satisfies  $f(x^*) = 1$  for the challenge attribute  $x^*$ , and  $[\mathbf{r}_f, \mathbf{r}'_f]$  is generated as  $\mathbf{r}'_f \leftarrow \{0, 1\}^N$  and  $\mathbf{r}_f \leftarrow \mathbf{A}_\tau^{-1}(-\mathbf{v} - (\mathbf{B}_0 + \mathbf{B}_f)\mathbf{r}'_f)$ .

Let  $\mathbf{H} := \text{EvRelation}(f, x^*, \vec{\mathbf{B}}_{x^*})$ . Then it holds that  $\mathbf{B}_f - f(x^*)(\mathbf{I}_n \otimes \mathbf{g}^T) = (\vec{\mathbf{B}}_{x^*} - x^*\vec{\mathbf{G}})\mathbf{H}$ . From  $f(x^*) = 1$ , we have  $\mathbf{B}_f = \mathbf{A}\vec{\mathbf{R}}\mathbf{H} + (\mathbf{I}_n \otimes \mathbf{g}^T)$ . Hence we have  $[\mathbf{A}, \mathbf{B}_0 + \mathbf{B}_f] = [\mathbf{A}, \mathbf{A}(\mathbf{R}_0 + \vec{\mathbf{R}}\mathbf{H}) + (\mathbf{I}_n \otimes \mathbf{g}^T)]$ . By Corollary 2.2, when given  $\mathbf{R}_0, \vec{\mathbf{R}}$  and  $\mathbf{H}$ , for any  $\tau \geq \tau' = O(\sqrt{mn} \cdot N \cdot \|(\mathbf{R}_0 + \vec{\mathbf{R}}\mathbf{H})\|_\infty)$ , we can sample from  $[\mathbf{A}, \mathbf{B}_0 + \mathbf{B}_f]_P^{-1}$  for  $P = D_{\mathbb{Z}^m, \tau} \times \{0, 1\}^N$ .

We generate  $[\mathbf{r}_f, \mathbf{r}'_f]$  by  $[\mathbf{r}_f, \mathbf{r}'_f] \leftarrow [\mathbf{A}, \mathbf{B}_0 + \mathbf{B}_f]_P^{-1}(-\mathbf{v})$ . Then,  $\mathbf{r}'_f$  is stored as the reply for the random oracle query  $(\mathbf{A}, f)$ . By Corollary 2.2, the marginal distribution of  $\mathbf{r}'_f$  is statistically indistinguishable from uniform over  $\{0, 1\}^N$ , and the probability distribution of  $\mathbf{r}_f$  conditioned on  $\mathbf{r}'_f$  is a discrete Gaussian distribution over the appropriate coset of the integer lattice. Since the view of the adversary in this game is statistically indistinguishable from that of **Game<sub>3</sub>**, we have

$$|\text{Adv}_{\mathcal{A}}^{\text{Game}_4}(\lambda) - \text{Adv}_{\mathcal{A}}^{\text{Game}_3}(\lambda)| = \text{negl}(\lambda).$$

- **Game<sub>5</sub>**: This game is the same as **Game<sub>4</sub>** except for the way to choose  $\mathbf{A}$ . The challenger chooses random  $\mathbf{A}$  from  $\mathbb{Z}_q^{n \times m}$  instead of generating it by using **TrapGen**. By Corollary 2.1, the distribution of the matrix  $\mathbf{A}$  generated by **TrapGen** is statistically indistinguishable from uniform over  $\mathbb{Z}_q^{n \times m}$ , so we have

$$|\text{Adv}_{\mathcal{A}}^{\text{Game}_5}(\lambda) - \text{Adv}_{\mathcal{A}}^{\text{Game}_4}(\lambda)| = \text{negl}(\lambda).$$

- **Game<sub>6</sub>**: We change the contents of the challenge ciphertexts as follows:

$$\begin{aligned} \mathbf{u}_A^{(C)} &:= \mathbf{A}^T \mathbf{s} + \mathbf{e}_A, u_v^{(C)} := \mathbf{v}^T \mathbf{s} + e_v, & \mathbf{u}_A^{(F)} &:= \mathbf{A}^T \mathbf{r} + \mathbf{e}_A^{(F)}, \\ u_v^{(F)} &:= \mathbf{v}^T \mathbf{r} + e_v^{(F)}, \mathbf{u}_A^{(D,k)} := \mathbf{A}^T \mathbf{s}^{(k)} + \mathbf{e}_A^{(k)}, & u_v^{(D,k)} &:= \mathbf{v}^T \mathbf{s}^{(k)} + e_v^{(k)}. \end{aligned}$$

The challenge ciphertexts can be rewritten as

$$\begin{aligned} \mathbf{c} &:= \begin{bmatrix} \mathbf{u}_A^{(C)} \\ \mathbf{R}_0^T \mathbf{u}_A^{(C)} \\ u_v^{(C)} \end{bmatrix}, \mathbf{c}_{x^*} := \begin{bmatrix} \mathbf{R}_1^T \mathbf{u}_A^{(C)} \\ \vdots \\ \mathbf{R}_\ell^T \mathbf{u}_A^{(C)} \end{bmatrix}, & \mathbf{f} &:= \begin{bmatrix} \mathbf{u}_A^{(F)} \\ (\mathbf{R}^{(F)})^T \mathbf{u}_A^{(F)} \\ u_v^{(F)} \end{bmatrix}, \\ \mathbf{d}^{(k)} &:= \begin{bmatrix} \mathbf{u}_A^{(D,k)} \\ \mathbf{R}_0^T \mathbf{u}_A^{(D,k)} \\ u_v^{(D,k)} \end{bmatrix}, \mathbf{d}_{x^*}^{(k)} := \begin{bmatrix} \mathbf{R}_1^T \mathbf{u}_A^{(D,k)} \\ \vdots \\ \mathbf{R}_\ell^T \mathbf{u}_A^{(D,k)} \end{bmatrix} & (\forall k \in [N]). \end{aligned}$$

This game is equivalent to **Game<sub>5</sub>**, so we have

$$\text{Adv}_{\mathcal{A}}^{\text{Game}_6}(\lambda) = \text{Adv}_{\mathcal{A}}^{\text{Game}_5}(\lambda).$$



- Game<sub>7</sub>: We change the distribution of  $\mathbf{u}_A^{(C)}, u_v^{(C)}, \mathbf{u}_A^{(F)}, u_v^{(F)}, \mathbf{u}_A^{(D,k)}, u_v^{(D,k)}$  to the uniform distribution. By the DLWE <sub>$n, q, \chi$</sub>  assumption, this change cannot be distinguished by the adversary  $\mathcal{A}$  and so we have

$$|\text{Adv}_{\mathcal{A}}^{\text{Game}_7}(\lambda) - \text{Adv}_{\mathcal{A}}^{\text{Game}_6}(\lambda)| = \text{negl}(\lambda).$$

- Game<sub>8</sub>: In this game, we change the distribution of the challenge ciphertexts to the uniform. By the leftover hash lemma, the view of the adversary in this game is statistically indistinguishable from Game<sub>7</sub>, so we have

$$|\text{Adv}_{\mathcal{A}}^{\text{Game}_8}(\lambda) - \text{Adv}_{\mathcal{A}}^{\text{Game}_7}(\lambda)| = \text{negl}(\lambda).$$

The advantage of the adversary in this game is 0, that is,  $\text{Adv}_{\mathcal{A}}^{\text{Game}_8}(\lambda) = 0$ .

From the above sequences of the games, we can see that  $\text{Adv}_{\mathcal{A}}^{\text{column}}(\lambda) = \text{negl}(\lambda)$ , and therefore the proposed MT-HABE is selectively secure.  $\square$

## References

- [BCTW16] Zvika Brakerski, David Cash, Rotem Tsabary, and Hoeteck Wee. Targeted Homomorphic Attribute Based Encryption. In *TCC 2016-B*, pages 330–360, 2016.
- [BGG<sup>+</sup>14] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, and Vinod Vaikuntanathan. Fully Key-Homomorphic Encryption, Arithmetic Circuit ABE and Compact Garbled Circuit. In *EUROCRYPT*, pages 533–556, 2014.
- [BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) Fully Homomorphic Encryption without Bootstrapping. In *ITCS*, pages 309–325, 2012.
- [BLP<sup>+</sup>13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical Hardness of Learning with Errors. In *STOC*, pages 575–584, 2013.
- [Bra12] Zvika Brakerski. Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP. In *CRYPTO*, pages 868–886, 2012.
- [BV11a] Zvika Brakerski and Vinod Vaikuntanathan. Efficient Fully Homomorphic Encryption from (Standard) LWE. In *FOCS*, pages 97–106, 2011.
- [BV11b] Zvika Brakerski and Vinod Vaikuntanathan. Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages. In *CRYPTO*, pages 505–524, 2011.
- [BV16] Zvika Brakerski and Vinod Vaikuntanathan. Circuit-ABE from LWE: Unbounded Attributes and Semi-Adaptive Security. In *CRYPTO*, pages 363–384, 2016.
- [CLT14] Jean-Sébastien Coron, Tancrede Lepoint, and Mehdi Tibouchi. Scale-Invariant Fully Homomorphic Encryption over the Integers. In *PKC*, pages 311–328, 2014.
- [CM15] Michael Clear and Ciarán McGoldrick. Multi-Identity and Multi-Key Leveled FHE from Learning with Errors. In *CRYPTO*, pages 630–656, 2015.

- [CM16] Michael Clear and Ciarán McGoldrick. Attribute-Based Encryption Fully Homomorphic Encryption with a Bounded Number of Inputs. In *AFRICACRYPT*, pages 307–324, 2016.
- [DGHV10] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully Homomorphic Encryption over the Integers. In *EUROCRYPT*, pages 24–43, 2010.
- [Gen09a] Craig Gentry. *A FULLY HOMOMORPHIC ENCRYPTION SCHEME*. PhD thesis, Stanford University, Available at <http://crypto.stanford.edu/craig>, 2009.
- [Gen09b] Craig Gentry. Fully Homomorphic Encryption using Ideal Lattices. In *STOC*, pages 169–178, 2009.
- [GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. How to Use a Short Basis: Trapdoors for Hard Lattices and New Cryptographic Constructions. In *STOC*, pages 197–206, 2008.
- [GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based. In *CRYPTO*, pages 75–92, 2013.
- [GVW13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based Encryption for Circuits. In *STOC*, pages 545–554, 2013.
- [LTV12] Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-Fly Multiparty Computation on the Cloud via Multikey Fully Homomorphic Encryption. In *STOC*, pages 1219–1234, 2012.
- [LW15] Vadim Lyubashevsky and Daniel Wichs. Simple Lattice Trapdoor Sampling from a Broad Class of Distributions. In *PKC*, pages 716–730, 2015.
- [MP12] Daniele Micciancio and Chris Peikert. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In *EUROCRYPT*, pages 700–718, 2012.
- [MW16] Pratyay Mukherjee and Daniele Wichs. Two-Round Multiparty Computation via Multi-Key FHE. In *EUROCRYPT*, pages 735–763, 2016.
- [Pei09] Chris Peikert. Public-Key Cryptosystems from the Worst-Case Shortest Vector Problem. In *STOC*, pages 333–342, 2009.
- [PS16] Chris Peikert and Sina Shiehian. Multi-Key FHE from LWE, Revisited. In *TCC 2016-B*, pages 217–238, 2016.
- [Reg05] Oded Regev. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. In *STOC*, pages 84–93, 2005.
- [Sch87] Clause-Peter Schnorr. A Hierarchy of Polynomial Time Lattice Basis Reduction Algorithm. *Theoretical Computer Science*, 53(2-3):201–224, 1987.