Dynamic Multi Target Homomorphic Attribute-Based Encryption

Ryo Hiromasa(✉) and Yutaka Kawai

Mitsubishi Electric, Kamakura, Japan
Hiromasa.Ryo@ajMitsubishiElectric.co.jp,
Kawai.Yutaka@daMitsubishiElectric.co.jp

Abstract. We propose multi target homomorphic attribute-based encryption (MT-HABE) with *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 dynamic homomorphic evaluations, we construct MT-HABE from a dual variant of multi-hop MKFHE proposed by Peikert and Shiehian (TCC 2016-B).

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 mpk is used to generate a ciphertext of a message μ, which is labeled with a public attribute $x \in \{0,1\}^\ell$. The secret key sk_f is associated to a policy $f : \{0,1\}^\ell \to \{0,1\}$ 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+14, BV16], and it was known that from [GSW13, 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.

M. O'Neill (Ed.): IMACC 2017, LNCS 10655, pp. 25–43, 2017.
https://doi.org/10.1007/978-3-319-71045-7_2

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 $\{(\mathsf{ct}_j, x_j)\}$, where each ciphertext encrypts μ_j. If for any x_j there exists f_i such that $f_i(x_j) = 0$, the algorithm outputs a ciphertext $\mathsf{ct}_F^{(g)}$ that can be decrypted by using all of the secret keys $\{\mathsf{sk}_{f_i}\}_i$, 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 *static* (i.e., single-hop for policies): 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 construct *dynamic* MT-HABE (i.e., which is multi-hop for policies): 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 encrypted under other keys. To implement dynamic homomorphic evaluation algorithms, we construct MT-HABE from a dual variant of 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 tells that the size of the public key of our scheme is almost the same as [BCTW16] ignoring the logarithmic factor.

1.2 Our Techniques

For the notation of this section, we refer the reader to the first paragraph of Sect. 2. 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 $\mathbf{g}^T := (1, 2, 2^2, \ldots, 2^{\lceil \log q \rceil})$. In the following, we use the notation $x \approx y$ to represent

Table 1. Comparison of key and ciphertext size between the previous MT-HABE [BCTW16] and our dynamic MT-HABE. The parameter n is the LWE dimension, ℓ is the maximal number of inputs of policies, $d_{\mathsf{BCTW}} = d_{\mathcal{F}} + d_{\mathcal{G}} \log d$, and $d_{\mathsf{ours}} = d \log d + d_{\mathcal{G}} + d_{\mathcal{F}} \log \ell$, where d represents the bound on the number of involved 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 $\to$ represent the size of a fresh and evaluated ciphertext.

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

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 $(\mathbf{C}, \mathbf{F}, \mathbf{D}) \in \mathbb{Z}_q^{n \times N} \times \mathbb{Z}_q^{n \times N} \times \mathbb{Z}_q^{nm \times N \lceil \log q \rceil}$ such that for a secret key vector $\mathbf{t} \in \mathbb{Z}_q^n$,

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

where $\hat{\mathbf{F}} = \mathbf{AR} \in \mathbb{Z}_q^{m \times N}$ for a random public matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a random binary matrix $\mathbf{R} \in \mathbb{Z}_q^{m \times N}$. To achieve dynamic homomorphism on the ciphertexts, the MKFHE has an algorithm to expand a ciphertext $\mathbf{C}$ under $\mathbf{t}$ into a ciphertext $\mathbf{C}' \in \mathbb{Z}_q^{(n+n) \times (n+n) \lceil \log q \rceil}$ under $\mathbf{t}' = [\mathbf{t}, \mathbf{t}^*] \in \mathbb{Z}_q^{n+n}$ for an additional key $\mathbf{t}^* \in \mathbb{Z}_q^n$, 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 $\mathbf{X} \in \mathbb{Z}_q^{n \times N}$ that satisfies $\mathbf{t}^T \mathbf{X} + \mathbf{t}^T \mathbf{AR} \approx \mathbf{0}$. 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 $\mathbf{A}, \mathbf{B}_0, \mathbf{B}_1, \ldots, \mathbf{B}_\ell \in \mathbb{Z}_q^{n \times m}$ and a random vector $\mathbf{v} \in \mathbb{Z}_q^n$. We define $\mathbf{B}_x := [\mathbf{B}_1, \ldots, \mathbf{B}_\ell]$, and $x\mathbf{G} := [x_1(\mathbf{I}_n \otimes \mathbf{g}^T), \ldots, x_\ell(\mathbf{I}_n \otimes \mathbf{g}^T)]$ for an attribute $x \in \{0,1\}^\ell$. A ciphertext of the ST-HABE consists of the following two matrices

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

$$\mathbf{C}_x \approx (\mathbf{B}_x - x\mathbf{G})^T \cdot \mathbf{S} \in \mathbb{Z}_q^{\ell N \times M}$$

for some random matrix $\mathbf{S}$. Let $f : \{0,1\}^\ell \to \{0,1\}$ 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 $\mathbf{B}_f - f(x)(\mathbf{I}_n \otimes \mathbf{g}^T) = (\mathbf{B}_x -$

$x\mathbf{G})\mathbf{H}$. 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 dynamic MT-HABE by making the multi-hop MKFHE [PS16] attribute-based. To this end, we consider a dual variant of [PS16]: set $\mathbf{C}$ in the same way as [BCTW16] and $\hat{\mathbf{F}} = [\mathbf{A}, \mathbf{B}, \mathbf{v}]^T\mathbf{R} + \mathbf{E} \approx [\mathbf{A}, \mathbf{B}, \mathbf{v}]^T\mathbf{R} \in \mathbb{Z}_q^{(m+N+1)\times M}$ for a random matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times N}$. In the scheme of [PS16], the matrix $\mathbf{F}$ contains a message μ, so $\hat{\mathbf{F}}$ must be indistinguishable from uniform to ensure the security. The matrix $\hat{\mathbf{F}}$ of [PS16] is set to be $\mathbf{AR}$, and so statistically indistinguishable from uniform by the leftover hash lemma (LHL). In our scheme, $\hat{\mathbf{F}}$ 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

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

1.3 Organization

In Sect. 2, we introduce mathematical preliminaries used in this paper. In Sect. 3, we show the construction of the proposed dynamic MT-HABE.

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 $\{1, 2, \ldots, d\}$ 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 $\mathsf{negl}(\lambda)$ 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 ℓ_∞ 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 ℓ_∞ 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 says

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

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

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

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,

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

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 here define the syntax of T-HABE and then define its correctness and security.

Definition 1 (Target Homomorphic Attribute Based Encryption (T-HABE)). *A target homomorphic attribute based encryption scheme consists of the following five algorithms* $\mathsf{THABE} = \mathsf{THABE}.\{\mathsf{Setup}, \mathsf{Enc}, \mathsf{Keygen}, \mathsf{Dec}, \mathsf{EvalNAND}\}$.

- $\mathsf{THABE.Setup}(1^\lambda)$*: takes as input a security parameter* λ *(additionally, the algorithm can take parameters that specify classes of policies or admissible operations), and outputs a public parameter* pp *and master secret key* msk.
- $\mathsf{THABE.Enc}_{\mathsf{pp}}(\mu, x)$*: takes as input a public parameter* pp*, plaintext* μ*, and attribute* x*, and outputs a tuple of a ciphertext and attribute* (ct, x).
- $\mathsf{THABE.Keygen}_{\mathsf{msk}}(f)$*: takes as input a master secret key* msk *and policy* f*, and outputs a secret key* sk_f.
- $\mathsf{THABE.EvalNAND}_{\mathsf{pp}}(\mathsf{ct}^{(1)}, \mathsf{ct}^{(2)})$*: takes as input a public parameter* pp*, and two ciphertexts* $\mathsf{ct}^{(1)}, \mathsf{ct}^{(2)}$*, and outputs a ciphertext* $\mathsf{ct}^{\mathsf{NAND}}$.
- $\mathsf{THABE.Dec}_{\mathsf{sk}_F}(\mathsf{ct})$*: takes as input a secret key* sk_F *(*$\mathsf{sk}_F = \{\mathsf{sk}_f : f \in F\}$*) and ciphertext* ct*, and outputs a plaintext* $\mu \in \{0, 1\}$.

As well as the definition of multi-hop MKFHE in [PS16], we consider the algorithm $\mathsf{EvalNAND}$, which homomorphically evaluates the NAND gate for two input ciphertexts, to capture multi-hop property in the above definition. The circuit evaluation algorithm $\mathsf{Eval}(\{\mathsf{ct}^{(i)}\}_i, \{f_j\}_j \subseteq \mathcal{F}, g \in \mathcal{G})$ takes as input an operation g composed of NAND gates of two inputs and one output, and computes each gate by $\mathsf{EvalNAND}$ on ciphertexts associated to the inputs of the gate. Each gate evaluation depends on the policies related only with the input ciphertexts, which makes the homomorphic evaluation multi-hop for policies.

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 involved in the homomorphic computation.

Definition 2 (Correctness). *Let* $\{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ *be a class of policies, and* $\{\mathcal{G}_\lambda\}_{\lambda\in\mathbb{N}}$ *be a class of operations. The dynamic T-HABE scheme* $\mathsf{THABE} = \mathsf{THABE}.\{\mathsf{Setup}, \mathsf{Enc}, \mathsf{Keygen}, \mathsf{Eval}, \mathsf{Dec}\}$ *is correct if the following holds.*

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

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

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 3 (Security). *Let* 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* $(\mathsf{msk}, \mathsf{pp}) \leftarrow \mathsf{THABE.Setup}(1^\lambda)$ *and sends* 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* $\mathsf{sk}_{f_i} \leftarrow \mathsf{THABE.Keygen}_{\mathsf{msk}}(f_i)$ *and sends* sk_{f_i} *if* $f_i(x^*) = 1$*, and sends* $\perp$ *otherwise.*

4. *The adversary sends a pair of messages* μ_0, μ_1 *to the challenger. The challenger chooses* $b \leftarrow \{0,1\}$ *uniformly at random, and computes* $\mathsf{ct}^* \leftarrow \mathsf{THABE.Enc}_{\mathsf{pp}}(\mu_b, x^*)$. *It sends* 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 $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{SS-THABE}}(\lambda) := |\Pr[b' = b] - 1/2|$, *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* $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{SS-THABE}}(\lambda) = \mathsf{negl}(\lambda)$.

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 4 (DLWE). *For a security parameter* λ, *let* $n := n(\lambda)$ *be an integer lattice dimension,* $q := q(\lambda) \geq 2$ *be an integer modulus, and* $\chi := \chi(\lambda)$ *be an error distribution over* $\mathbb{Z}$. $\mathsf{DLWE}_{n,q,\chi}$ *is the problem that for any* $m = \mathsf{poly}(\lambda)$, *letting* $\mathbf{A} \leftarrow \mathbb{Z}_q^{m \times n}$, $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow \chi^m$, *and* $\mathbf{u} \leftarrow \mathbb{Z}_q^m$, *distinguishes the two distributions* $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ *and* $(\mathbf{A}, \mathbf{u})$. $\mathsf{DLWE}_{n,q,\chi}$ *assumption states that* $\mathsf{DLWE}_{n,q,\chi}$ *is intractable for any PPT adversary.*

By letting χ 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] between $\mathsf{DLWE}_{n,q,\chi=D_{\mathbb{Z},r}}$ and approximating a short vector over n dimensional lattices within factor of $\tilde{O}(n/\alpha)$[1]. Additionally, it is known that there exists the classical reductions [Pei09, BLP+13] for other parameters.

2.3 Gadget Matrix and Bit Decomposition

Let $\mathbf{g}^T := (1, 2, \ldots, 2^{\lceil \log q \rceil})$ be a vector consisting of the powers of 2. The operation $\mathbf{g}^{-1} : \mathbb{Z}_q \to \{0,1\}^{1 \times \lceil \log q \rceil}$ 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, $\mathbf{g}^{-T} : \mathbb{Z}_q \to \{0,1\}^{\lceil \log q \rceil}$ 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.

[1] Approximating a short vector over n dimensional lattices within factor of γ takes $2^{\tilde{\Omega}(n/\log \gamma)}$ 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 τ.

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

Corollary 1 (Generating Trapdoors [Ajt99,GPV08,MP12,BLP+13]). *There exists an efficient algorithm* $\mathsf{TrapGen}(1^n, q, m)$ *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 (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* $[\mathbf{A}, \mathbf{AR} + (\mathbf{I}_n \otimes \mathbf{g}^T)]_P^{-1}$ *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* $[\mathbf{A}, \mathbf{AR} + (\mathbf{I}_n \otimes \mathbf{g}^T)]_P^{-1}(\mathbf{v})$ *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 5. *Let* $n, q, \ell \in \mathbb{N}$ *and* $N := n\lceil\log q\rceil$*. Consider* $\mathbf{B}_1, \ldots, \mathbf{B}_\ell \in \mathbb{Z}_q^{n\times N}$*, and denote* $\mathbf{B} := [\mathbf{B}_1, \ldots, \mathbf{B}_\ell]$*. Let* f *be a Boolean circuit of depth* d *that computes a function* $\{0,1\}^\ell \to \{0,1\}$ *and consists only of NAND gates. We define* $\mathbf{B}_f := \mathsf{Eval}(\mathbf{B}, f)$ *recursively: associate* $\mathbf{B}_1, \ldots, \mathbf{B}_\ell$ *with the* ℓ *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 21. *Consider* $\mathbf{B}_1, \ldots, \mathbf{B}_\ell \in \mathbb{Z}_q^{n\times N}$ *(*$N = n\lceil\log q\rceil$*). Letting* $\mathbf{B} := [\mathbf{B}_1, \ldots, \mathbf{B}_\ell]$*, and* $x\mathbf{G} := [x_1(\mathbf{I}_n \otimes \mathbf{g}^T), \ldots, x_\ell(\mathbf{I}_n \otimes \mathbf{g}^T)]$*, there exists a polynomial time algorithm* $\mathsf{EvRelation}$ *such that if* $\mathbf{H} := \mathbf{H}_{f,x,\mathbf{B}} := \mathsf{EvRelation}(f, x, \mathbf{B})$*, then* $\|\mathbf{H}\|_\infty \leq (N+1)^d$ *and*

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

where $\mathbf{B}_f = \mathsf{Eval}(f, \mathbf{B})$*.*

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

We can see that this fact holds by verifying that for the NAND operation in Definition 5,

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

3 Dynamic MT-HABE

In this section, we construct dynamic MT-HABE (i.e., which is multi-hop for keys) 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 Sect. 3.1 except for homomorphic evaluation algorithms, which are described in Sect. 3.2.

3.1 Construction

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

- $\mathsf{dMTHABE.Setup}(1^\lambda, 1^\ell, 1^{d_\mathcal{F}}, 1^{d_\mathcal{G}}, 1^d)$: choose DLWE parameters n, q, χ as described in Appendix A.1. Let B be a bound of samples from error distribution χ. Let $m = O(n \log^2 q)$, $N := n\lceil \log q \rceil$, and $M := (m + N + 1)\lceil \log q \rceil$. Generate $(\mathbf{A}, \mathbf{A}_{\tau_0}^{-1}) \leftarrow \mathsf{TrapGen}(1^n, q, m)$, where $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\tau_0 = O(\sqrt{n \log q \log n})$ from Corollary 1. Sample random matrices $\mathbf{B}, \mathbf{B}_0, \mathbf{B}_1, \ldots, \mathbf{B}_\ell \leftarrow \mathbb{Z}_q^{n\times N}$, and let $\mathbf{B}_x := [\mathbf{B}_1, \ldots, \mathbf{B}_\ell]$. Sample a random vector $\mathbf{v} \leftarrow \mathbb{Z}_q^n$. Choose a PRF seed $\sigma \leftarrow \mathsf{PRF.Gen}(1^\lambda)$. Let $H : \mathbb{Z}_q^{n\times m} \times \mathcal{F} \to \{0,1\}^N$ be a hash function implemented by the random oracle. Output $\mathsf{pp} := (\mathbf{A}, \mathbf{B}, \mathbf{B}_0, \mathbf{B}_x, \mathbf{v}, H)$ and $\mathsf{msk} := (\mathbf{A}_{\tau_0}^{-1}, \sigma)$.
- $\mathsf{dMTHABE.Enc}_{\mathsf{pp}}(\mu \in \{0,1\}, x \in \{0,1\}^\ell)$: 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 $i \in \{0, 1, \ldots, \ell\}$ 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

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

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

  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

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

For every $i \in \{0, 1, \ldots, \ell\}$, $j \in [M]$, 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 $\mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(N)} \leftarrow \chi^{n\times M}$, and compute

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

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

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

– $\mathsf{dMTHABE.Keygen}_{\mathsf{msk}}(f \in \mathcal{F})$: compute $\mathbf{B}_f := \mathsf{Eval}(f, \mathbf{B}_x)$ from f and $\mathbf{B}_x$. 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 $\rho \leftarrow \mathsf{PRF.Eval}(\sigma, f)$, 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 $\mathsf{sk}_f := \mathbf{r}_f$.

– $\mathsf{dMTHABE.ApplyF}_{\mathsf{pp}}(\mathsf{ct}, f \in \mathcal{F})$: when given ct and f, first compute a matrix $\mathbf{H} := \mathsf{EvRelation}(f, x, \mathbf{B}_x)$. Then set $\mathbf{C}_f := \mathbf{H}^T \mathbf{C}_x$, and compute

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

By Fact 21, it holds that for a secret key $\mathbf{t}_f^T := [\mathbf{r}_f^T, \mathbf{r}_f'^T, 1]$ with related to f,

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

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

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

Compute $\hat{\mathbf{D}}_f := \mathbf{D} + \mathbf{D}_f$. Similar to $\hat{\mathbf{C}}_f$, 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 $\mathsf{ct}^{(f)} := (\hat{\mathbf{C}}_f, \mathbf{F}, \hat{\mathbf{D}}_f)$.

- $\mathsf{dMTHABE.Eval}(\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(k)}, F := \{f_1, \ldots, f_d\} \subseteq \mathcal{F}, g \in \mathcal{G})$: for fresh ciphertext $\mathsf{ct}^{(i)}$, compute functioned ciphertext $\mathsf{ct}^{(f_i)} := \mathsf{dMTHABE.ApplyF}_{\mathsf{pp}}(\mathsf{ct}^{(i)}, f_j)$, where $f_j \in F$ such that $f_j(x_i) = 0$. Then homomorphically evaluate g between the ciphertexts, and output $\mathsf{ct}^{(F)}$.
- $\mathsf{dMTHABE.Dec}_{\mathsf{sk}_{f_1}, \ldots, \mathsf{sk}_{f_d}}(\mathsf{ct}^{(F)})$[2]: given secret keys $\mathsf{sk}_{f_1}, \ldots, \mathsf{sk}_{f_d}$ for every policy in $F = \{f_1, \ldots, f_d\}$, and an ciphertext $\mathsf{ct}^{(F)} = (\hat{\mathbf{C}}_F, \mathbf{F}, \hat{\mathbf{D}}_F)$ 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 $\mathbf{t}_F^T := [\mathbf{r}_{f_1}^T, \mathbf{r}'^T_{f_1}, 1, \ldots, \mathbf{r}_{f_d}^T, \mathbf{r}'^T_{f_d}, 1]$, and compute a vector $\mathbf{c} := \mathbf{t}_F^T \hat{\mathbf{C}}_F$. Let $\mathbf{u}^T := (0, \ldots, 0, \lfloor q/2 \rfloor) \in \mathbb{Z}^{1 \times d(m+N+1)}$. 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 A.

3.2 The Algorithm Eval

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

Suppose that we obtain a functioned ciphertext $\mathsf{ct}^{(f)} := (\hat{\mathbf{C}}_f, \mathbf{F}, \hat{\mathbf{D}}_f)$ by applying dMTHABE.ApplyF for a policy $f \in \mathcal{F}$ to a fresh ciphertext $\mathsf{ct} := (x, \mathbf{C}, \mathbf{C}_x, \mathbf{F}, \mathbf{D}, \{\mathbf{D}_x^{(k)}\}_k)$. Then the functioned ciphertext $\mathsf{ct}^{(f)}$ 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.

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

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

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

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 $\mathbf{t} \in \mathbf{Z}_q^{n'}$ ($n' = k(m+N+1)$ 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 $\tilde{\mathbf{t}} := [\mathbf{t}, \mathbf{t}_f]$ constructed from $\mathbf{t}$ and $\mathbf{t}_f := [\mathbf{r}_f^T, \mathbf{r}'^T_f, 1]^T \in \mathbb{Z}_q^{m+N+1}$, and random matrix $\tilde{\mathbf{R}}$:

[2] The algorithm can take as input fresh ciphertext ct (and the single secret key sk_f for $f \in \mathcal{F}$ such that $f(x) = 0$) by generating the functioned ciphertext $\mathsf{ct}^{(F)} := \mathsf{dMTHABE.ApplyF}_{\mathsf{pp}}(\mathsf{ct}, f)$ before the computation begins.

– $\mathbf{F}$ 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.

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

– 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 $\mathbf{B}_f := \mathsf{Eval}(\mathbf{B}, f)$. 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

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

From

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

we have

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

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 $B_{\mathbf{C}_i}$ be a upper bound of the noise included in $\mathbf{C}_i (i = 1, 2)$, then we have

$$\begin{aligned}
\mathbf{t}^T \mathbf{C}_{\text{mult}} &\approx \mu_1 (\mathbf{t}^T \otimes \mathbf{g}^T) \cdot (\mathbf{I}_{n'} \otimes \mathbf{g}^{-T})(\mathbf{C}_2) \quad (\text{error: } n' \lceil \log q \rceil B_{\mathbf{C}_1}) \\
&= \mu_1 \mathbf{t}^T \mathbf{C}_2 \\
&\approx \mu_1 \mu_2 (\mathbf{t}^T \otimes \mathbf{g}^T) \qquad (\text{error: } \mu_1 B_{\mathbf{C}_2}).
\end{aligned}$$

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

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

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

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

In addition, the following holds:

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

Hence, by

$$(\mathbf{I}_N \otimes \mathbf{t}^T)\mathbf{D}_{\text{mult}} \approx \mathbf{R}_{\text{mult}} \otimes \mathbf{g} \quad (\text{error: } M \cdot B_{\mathbf{D}_1} + n'\lceil \log q \rceil B_{\mathbf{C}_1} + \mu_1 B_{\mathbf{D}_2}),$$

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

A Correctness and Security

In this section, we discuss about correctness and security of the proposed MT-HABE described in Sect. 3. In Appendix A.1, we consider parameter settings of the proposed scheme for the correctness and security, and the proofs of them are described in Appendix A.2.

A.1 Parameter Settings

The DLWE parameters n, q, χ 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 $\ell, d = \mathsf{poly}(\lambda)$. 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 Sect. 3.2, the ciphertext generated by homomorphically evaluating a NAND gate has noise at most

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

for some polynomial $\mathsf{poly}(\cdot)$. The ciphertext generated by the ciphertext expansion algorithm described in Sect. 3.2 also has noise at most

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

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

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

$$\begin{aligned}&\mathsf{poly}(d, n, \lceil \log q \rceil)^d \cdot \mathsf{poly}'(n, \lceil \log q \rceil)^{d_\mathcal{G}} \cdot \|\mathbf{t}_f\|_\infty \cdot ((N+1)^{d_\mathcal{F}} \cdot \ell N + 1)mB \\ &\leq \mathsf{poly}(d, n, \lceil \log q \rceil)^d \cdot \mathsf{poly}'(n, \lceil \log q \rceil)^{d_\mathcal{G}} \cdot O(\ell^2 m^2 \sqrt{n} N^3 (N+1)^{2d_\mathcal{F}})B.\end{aligned}$$

For the correctness and security, we select the parameters so that the above quantity by a factor of eight is less than 2^{n^ϵ} 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 χ so that they satisfy $q/B \geq 2^{n^\epsilon}$, where B is the upper bound of the noise distribution χ. Selecting such parameters leads the reduction from the $\mathsf{DLWE}_{n,q,\chi}$ problem to approximate a short vector on the n dimensional lattice by a factor of $\tilde{O}(n \cdot 2^{n^\epsilon})$.

A.2 Proofs

Correctness and security of our dMTHABE scheme can be proven in a very similar way to [BCTW16].

Theorem 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. Let $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{dMTHABE.Setup}(1^\lambda, 1^\ell, 1^{d_\mathcal{F}}, 1^{d_\mathcal{G}}, 1^d)$. Consider k ciphertexts $\mathsf{ct}^{(i)} \leftarrow \mathsf{dMTHABE.Enc}_{\mathsf{pp}}(\mu_i, x_i)$ of message $\mu_i \in \{0,1\}$ with attribute $x_i \in \{0,1\}^\ell$. For a set of d policies $F := \{f_i\}_{i\in[d]} \subseteq \mathcal{F}_{\ell,d_\mathcal{F}}$ and operation $g \in \mathcal{G}_{d_\mathcal{G}}$, consider an evaluated ciphertext

$$\mathsf{ct}^{(F)} := (\hat{\mathbf{C}}_F, \mathbf{F}_F, \hat{\mathbf{D}}_F) := \mathsf{dMTHABE.Eval}(\{\mathsf{ct}^{(i)}\}_{i\in[k]}, F, g).$$

By the process of Eval in Sect. 3.2, it holds that

$$\mathbf{c} := \mathbf{t}_F \hat{\mathbf{C}}_F \approx \mu_g(\mathbf{t}_F^T \otimes \mathbf{g}^T)$$

for $\mu_g := g(\mu_1, \ldots, \mu_k)$ and $\mathbf{t}_F^T := [\mathbf{t}_{f_1}^T, \ldots, \mathbf{t}_{f_d}^T]$ where $\mathbf{r}_{f_i} \leftarrow \mathsf{dMTHABE.Keygen}_{\mathsf{msk}}(f_i)$, $\mathbf{r}'_{f_i} = H(\mathbf{A}, f_i)$, and $\mathbf{t}_{f_i}^T := [\mathbf{r}_{f_i}^T, \mathbf{r}'^T_{f_i}, 1]$. Let $\mathbf{u}^T := (0, \ldots, 0, \lfloor q/2 \rfloor)$, then

$$\tilde{\mu} := \mathbf{c}^T(\mathbf{I}_{d(m+N+1)} \otimes \mathbf{g}^{-T})(\mathbf{u}) \approx \mu_g \lfloor q/2 \rfloor.$$

Choosing the parameters as described in Appendix A.1, the noise in $\hat{\mathbf{C}}_F$ is of size at most $q/8$. Hence, it holds that

$$\Pr[\mathsf{dMTHABE.Dec}_{\mathsf{sk}_{f_1},\ldots,\mathsf{sk}_{f_d}}(\mathsf{ct}^{(F)}) \neq \mu_g] = \mathsf{negl}(\lambda).$$

Theorem 2 (Security). *The scheme* dMTHABE *scheme is selectively secure for function classes $\mathcal{F}, \mathcal{G}$ in the random oracle model if the* $\mathsf{DLWE}_{n,q,\chi}$ *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

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

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

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 $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{column}}(\lambda)$. Without loss of generality, we can prove the security in the column game instead of proving the selective security game defined in Definition 3.

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

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

– 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 $\mathsf{negl}(\lambda)$, we have

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

– 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:

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

– Game_3: This game is the same as Game_2 except for the generation of the public parameters $\mathbf{B}, \mathbf{B}_0, \mathbf{B}_1, \ldots, \mathbf{B}_\ell$. Here, there exist matrices $\mathbf{R}_0, \mathbf{R}_1, \ldots, \mathbf{R}_\ell$ 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 $\mathbf{B}, \mathbf{B}_0, \mathbf{B}_1, \ldots, \mathbf{B}_\ell$ 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 $\mathbf{B}, \mathbf{B}_0, \mathbf{B}_1, \ldots, \mathbf{B}_\ell$ is indistinguishable from uniform over $\mathbb{Z}_q^{n\times N}$. Hence we have

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

– Game_4: This game is the same as Game_3 except that the return 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 $\mathbf{H} := \mathsf{EvRelation}(f, x^*, \mathbf{B}_{x^*})$. Then it holds that $\mathbf{B}_f - f(x^*)(\mathbf{I}_n \otimes \mathbf{g}^T) = (\mathbf{B}_{x^*} - x^*\mathbf{G})\mathbf{H}$. From $f(x^*) = 1$, we have $\mathbf{B}_f = \mathbf{ARH} + (\mathbf{I}_n \otimes \mathbf{g}^T)$. Hence we have $[\mathbf{A}, \mathbf{B}_0 + \mathbf{B}_f] = [\mathbf{A}, \mathbf{A}(\mathbf{R}_0 + \mathbf{RH}) + (\mathbf{I}_n \otimes \mathbf{g}^T)]$. By Corollary 2, when given $\mathbf{R}_0$, $\mathbf{R}$ and $\mathbf{H}$, for any $\tau \geq \tau' = O(\sqrt{mn} \cdot N \cdot \|(\mathbf{R}_0 + \mathbf{RH})\|_\infty)$, 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, 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_3, we have

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

– Game_5: This game is the same as Game_4 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 $\mathsf{TrapGen}$. By Corollary 1, the distribution of the matrix $\mathbf{A}$ generated by $\mathsf{TrapGen}$ is statistically indistinguishable from uniform over $\mathbb{Z}_q^{n\times m}$, so we have

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

– Game_6: We change the contents of the challenge ciphertexts as follows:

$$\mathbf{u}_A^{(C)} := \mathbf{A}^T\mathbf{s} + \mathbf{e}_A, u_v^{(C)} := \mathbf{v}^T\mathbf{s} + e_v, \quad \mathbf{u}_A^{(F)} := \mathbf{A}^T\mathbf{r} + \mathbf{e}_A^{(F)},$$

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

The challenge ciphertexts can be rewritten as

$$\mathbf{c} := \begin{bmatrix} \mathbf{u}_A^{(C)} \\ \mathbf{R}_0^T\mathbf{u}_A^{(C)} \\ u_v^{(C)} \end{bmatrix}, \mathbf{c}_{x^*} := \begin{bmatrix} \mathbf{R}_1^T\mathbf{u}_A^{(C)} \\ \vdots \\ \mathbf{R}_\ell^T\mathbf{u}_A^{(C)} \end{bmatrix}, \quad \mathbf{f} := \begin{bmatrix} \mathbf{u}_A^{(F)} \\ (\mathbf{R}^{(F)})^T\mathbf{u}_A^{(F)} \\ u_v^{(F)} \end{bmatrix},$$

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

This game is equivalent to Game_5, so we have

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

– Game_7: 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 $\mathsf{DLWE}_{n,q,\chi}$ assumption, this change cannot be distinguished by the adversary $\mathcal{A}$ and so we have

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

– Game_8: 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_7, so we have

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

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

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

References

[Ajt99] Ajtai, M.: Generating hard instances of the short basis problem. In: ICALP, pp. 1–9 (1999)

[BCTW16] Brakerski, Z., Cash, D., Tsabary, R., Wee, H.: Targeted homomorphic attribute-based encryption. In: Hirt, M., Smith, A. (eds.) TCC 2016, Part II. LNCS, vol. 9986, pp. 330–360. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53644-5_13

[BGG+14] Boneh, D., Gentry, C., Gorbunov, S., Halevi, S., Nikolaenko, V., Segev, G., Vaikuntanathan, V., Vinayagamurthy, D.: Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 533–556. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55220-5_30

[BGV12] Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (leveled) Fully homomorphic encryption without bootstrapping. In: ITCS, pp. 309–325 (2012)

[BLP+13] Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: STOC, pp. 575–584 (2013)

[Bra12] Brakerski, Z.: Fully homomorphic encryption without modulus switching from classical GapSVP. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 868–886. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_50

[BV11a] Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. In: FOCS, pp. 97–106 (2011)

[BV11b] Brakerski, Z., Vaikuntanathan, V.: Fully homomorphic encryption from Ring-LWE and security for key dependent messages. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 505–524. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_29

[BV16] Brakerski, Z., Vaikuntanathan, V.: Circuit-ABE from LWE: unbounded attributes and semi-adaptive security. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part III. LNCS, vol. 9816, pp. 363–384. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53015-3_13

[CLT14] Coron, J.-S., Lepoint, T., Tibouchi, M.: Scale-Invariant Fully Homomorphic Encryption over the Integers. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 311–328. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54631-0_18

[CM15] Clear, M., McGoldrick, C.: Multi-identity and multi-key leveled FHE from learning with errors. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015, Part II. LNCS, vol. 9216, pp. 630–656. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48000-7_31

[CM16] Clear, M., McGoldrick, C.: Attribute-based fully homomorphic encryption with a bounded number of inputs. In: Pointcheval, D., Nitaj, A., Rachidi, T. (eds.) AFRICACRYPT 2016. LNCS, vol. 9646, pp. 307–324. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-31517-1_16

[DGHV10] van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully homomorphic encryption over the integers. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 24–43. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_2

[Gen09a] Gentry, C.: A fully homomorphic encryption scheme. Ph.D. thesis, Stanford University (2009). http://crypto.stanford.edu/craig

[Gen09b] Gentry, C.: Fully homomorphic encryption using ideal lattices. In: STOC, pp. 169–178 (2009)

[GPV08] Gentry, C., Peikert, C., Vaikuntanathan, V.: How to use a short basis: trapdoors for hard lattices and new cryptographic constructions. In: STOC, pp. 197–206 (2008)

[GSW13] Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 75–92. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40041-4_5

[GVW13] Gorbunov, S., Vaikuntanathan, V., Wee, H.: Attribute-based encryption for circuits. In: STOC, pp. 545–554 (2013)

[LTV12] López-Alt, A., Tromer, E., Vaikuntanathan, V.: On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In: STOC, pp. 1219–1234 (2012)

[LW15] Lyubashevsky, V., Wichs, D.: Simple lattice trapdoor sampling from a broad class of distributions. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 716–730. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46447-2_32

[MP12] Micciancio, D., Peikert, C.: Trapdoors for lattices: simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_41

[MW16] Mukherjee, P., Wichs, D.: Two round multiparty computation via multi-key FHE. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016, Part II. LNCS, vol. 9666, pp. 735–763. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_26

[Pei09] Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem. In: STOC, pp. 333–342 (2009)

[PS16] Peikert, C., Shiehian, S.: Multi-key FHE from LWE, revisited. In: Hirt, M., Smith, A. (eds.) TCC 2016, Part II. LNCS, vol. 9986, pp. 217–238. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53644-5_9

[Reg05] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC, pp. 84–93 (2005)

[Sch87] Schnorr, C.-P.: A hierarchy of polynomial time lattice basis reduction algorithm. Theor. Comput. Sci. **53**(2–3), 201–224 (1987)

http://www.springer.com/978-3-319-71044-0

Cryptography and Coding
16th IMA International Conference, IMACC 2017, Oxford, UK, December 12-14, 2017, Proceedings
O'Neill, M. (Ed.)
2017, X, 393 p. 23 illus., Softcover
ISBN: 978-3-319-71044-0