ATLANTIS PRESS

Advances in Computer Science Research, volume 62

2nd Joint International Information Technology, Mechanical and Electronic Engineering Conference (JIMEC 2017)

CCA-Secure Leveled FHE From Multi-Identity Fully Homomorphic Encryption

Weili Wang, Bin Hu, Xiufeng Zhao

Information Science and Technology Institute, Zhengzhou, 450001, China

Keywords: fully homomorphic encryption; CCA-secure; multi-identity fully homomorphic encryption; Lattice

Abstract

This paper proposes a generic construction of CCA-secure fully homomorphic encryption (FHE) scheme. First, we introduce a new primitive called multi-identity fully homomorphic encryption, which works in the multi-identity setting; that is, homomorphic evaluation can performed on ciphertexts created with different identities. Then, we propose an IND-sID-CPA secure multi-identity leveled FHE scheme over lattice. Finally, we present CCA-secure FHE from our proposed multi-identity FHE scheme and strongly EUF-CMA secure signature. The security analysis shows that our FHE scheme is CCA2-secure when the evaluation key is unavailable to the adversary, and remains CCA1-secure when the evaluation key is exposed in the standard model.

1 Introduction

With the rapid development of communication and information technology, cloud computing has become more and more popular. However, cloud security has gradually become a bottleneck restricting the development of cloud computing. With the appearance of fully homomorphic encryption (FHE), it is suitable for ensuring security in cloud environments.

Improvement in the security of FHE will lead to wider deployment of cloud-type applications. However, it is well known that adaptive chosen ciphertext (CCA2) security and the homomorphic property can never be achieved simultaneously[1]. In other words, security is sacrificed in exchange for the homomorphic property. So far, most FHE schemes satisfy CPA-secure. In present, constructing CCA1 secure FHE scheme is an interesting open problem.

The underlying cause of the incompatibility of CCA2 security and the homomorphic property lies in that every user can use the homomorphic property. But it is worth discussing whether the free availability of homomorphic operations is an indispensable functionality in real world applications. If everyone can perform a homomorphic operation, then it is hard to reduce the risk of unexpected changes to the encrypted data in the database in which resources are dynamically allocated. So we cannot rule out the possibility of unexpected changes to a user's data by any user who is authorized to access the database. Now we can see that the property that anyone can perform homomorphic operations not only inhibits the realization of CCA2 security, but also introduces the problem of unexpected modification of encrypted data. Setting access permissions on encrypted data has a practical significance.

In PKC 2013, Emura et al.[2] showed that CCA security does not rule out homomorphism when the capability to compute on encrypted data is controlled. Based on hash proof systems, Emura et al. constructed a number of CCA-secure keyed-homomorphic schemes. In EUROCRYPT 2014 Libert et al.[3] proposed quasi-adaptive noninteractive zero-knowledge proofs with unbounded simulation-soundness (USS), and constructed a CCA-secure keyed-homomorphic scheme with threshold decryption by applying USS. These two methods of constructing CCA-secure keyed homomor-phic schemes only allow simple computations on encrypted data, i.e., either adding or multiplying encrypted ciphertexts, but not both operations at the same time. In PKC 2016, Lai et al.[4] present a generic construction of CCA-secure keyed-FHE based on indistinguishable obfuscation[5], which is therefore highly inefficient at present time. So constructing realizable CCA-secure FHE scheme is still an open problem.

In EUROCRYPT 2004, Canetti et al.[6] proposed a simple and efficient construction of CCA-secure public key encryption (PKE) scheme from any CPA-secure identity-based encryption (IBE) scheme, called CHK transformation. They showed that combining an IND-sID-CPA secure IBE scheme with a strongly EUF-CMA secure signature scheme could get a CCA secure PKE scheme. In some sense, our work is inspired by CHK transformation.

1.1 Our results

We propose a CCA-secure FHE scheme based on the line of CHK transformation. First, we define a new primitive called multi-identity-based fully hommorphic encryption (IBFHE) and its IND-sID-CPA (indistin-guishable from random under a selective identity attack) security notions. Informally, a multi-identity IBFHE scheme is an identity-based fully homomorphic encryption scheme which works in the multi-identities setting. In other words, the scheme can evaluate on ciphertexts created with different identity. Based on the new primitive, we give a high-level description on how to construct a CCA-secure FHE scheme with the help of a strongly EUF-CPA-secure (existential unforgedable under adaptive chosen-message-attacks, [7]) signature scheme.

Generally speaking, the public key of our proposed FHE scheme is the public parameters of the multi-identity IBFHE scheme, the secret key is the corresponding master key, and the evaluation key is (vk', sk', evk), where (vk', sk') is a key-pair for the signature scheme $\mathcal{S}$, evk is generated by the multi-identity IBFHE scheme. A message bit is encrypted with respect to the "identity" vk, with the ciphertext denoted as CT.

The final ciphertexr is denoted as $C=(vk, CT, \sigma)$, where σ is a valid signature of *CT* by the signature key *sk*. For decrypting, the decryption algorithm should first verify the signature σ on *CT* with respect to *vk* and outputs $\perp$ if the verification fails. We will describe the approach in detail in part 3.

Finally, the security proof shows that the proposed FHE scheme is secure against chosen ciphertext attacks in the standard model.

1.2 Organization

This paper is organized as follows. In Section 2, we introduce definitions that we use throughout this paper including the definition of fully homomorphic encryption, and its CCA security definition. In Section 3, we present our construction of CCA secure FHE scheme, and prove the CCA security of the construction. Finally, we conclude the paper in Section 4.

2 Preliminaries

2.1 Lattice and LWE

Definition 1 (Lattice). Let $\boldsymbol{v}_i$ be linearly independent vectors in $\mathbb{Z}^m$. The *m*-dimensional full-rank lattice *L* is a linearly integer combination of these vectors:

$$L=\left\{\sum_{i=1}^{m} x_i v_i : x_i \in \mathbb{Z}, i=1,\cdots,m\right\}.$$

Definition 2 (*q*-ary Lattice). For *q* prime, $\boldsymbol{A}\in\mathbb{Z}_q^{n\times m}$ and $\boldsymbol{u}\in\mathbb{Z}_q^n$, define:

$$\Lambda^{\perp}(\boldsymbol{A})=\left\{\boldsymbol{y}\in\mathbb{Z}^m : \boldsymbol{A}\boldsymbol{y}=0 \bmod q\right\}$$

$$\Lambda^{\boldsymbol{u}}(\boldsymbol{A})=\left\{\boldsymbol{y}\in\mathbb{Z}^m : \boldsymbol{A}\boldsymbol{y}=\boldsymbol{u} \bmod q\right\}$$

Definition 3 (LWE). For an integer *n*, prime $q=q(n)$ and a distribution χ, the $\mathrm{LWE}_{n,q,\chi}$ problem is to distinguish the following two distributions: The one distribution is sampling (a_i, b_i) uniformly from $\mathbb{Z}_q^n\times\mathbb{Z}_q$. The other distribution is $(a_i, b_i=a_i\cdot s+e_i)\in\mathbb{Z}_q^n\times\mathbb{Z}_q$, where $s\in\mathbb{Z}_q^n$, $a_i\in\mathbb{Z}_q^n$ are drawn uniformly and e_i is an error term chosen from the noise distribution χ over $\mathbb{Z}$.

2.2 Fully Homomorphic Encryption

Definition 4 (Homomorphic Encryption). A homomorphic encryption scheme can be described as 4-tuple of algorithms $HE=(KeyGen, Enc, Dec, Eval)$ as follows:

- $KeyGen(1^n)$: On input the security parameter *n* and output (*PK*, *SK*, *EK*), where *PK* and *SK* are public key and secret key respectively, *EK* is the evaluation key.
- $Enc(PK, b)$: On input the public key *PK* and a single bit message $b\in\mathbb{Z}_2$, output a ciphertext *C*.
- $Dec(SK, C)$: On input the decryption key *SK* and a ciphertext *C*, and output a plaintext $b=Dec(SK, C)$.
- $Eval(EK, f, C_1, C_2, \cdots, C_k)$: On input the evaluation key *EK*, a function $f:\{0,1\}^k\to\{0,1\}$ and *k* ciphertexts $C_1, C_2, \cdots, C_k$, and output a ciphertext C_f.

Definition 5 (Correctness of FHE). A scheme FHE is correct if the following holds. For all (*PK*, *SK*, *EK*) output by $KeyGen(1^n)$, all message bit *b* and all arithmetic circuit *f*, with overwhelming probability we have:

(1) $Dec(SK, C_f)=f(Dec(SK, C_1), \cdots, Dec(SK, C_k))$.

(2) $Dec(SK, Enc(PK, b))=b$.

Definition 6 (*L*-Homomorphic). A *HE* scheme is *L*-Homomorphic if for arithmetic circuit $f:\{0,1\}^k\to\{0,1\}$ (over $GF(2)$) with depth no more than *L*, and respective inputs $b_1, b_2, \cdots, b_k\in\mathbb{Z}_2$, it holds that:

$$Pr[Dec_{SK}(Eval_{EK}(f, C_1, C_2, \cdots, C_k))\neq f(b_1, b_2, \cdots, b_k)]=negl(n)$$

where $(PK, SK, EK)\leftarrow KeyGen(1^n)$ and $C_i=Enc_{PK}(b_i)$.

2.3 CCA-security of FHE

The CCA security of FHE scheme is defined using the following game between a probabilistic polynomial time adversary $\mathcal{A}$ and a challenger. The adversary is only allowed to issue the decryption queries before it requests the evaluation key *EK* to be exposed in our security definition; thus it is slightly different from the definition given in [2]. That is, in our model, a FHE scheme should provide CCA security when the evaluation key is unavailable to the adversary and remain CPA-secure when the evaluation key is exposed.

Definition 7 (CCA-security of FHE). A fully homomor-phic encryption scheme *FHE*=(*KeyGen*, *Enc*, *Dec*, *Eval*) is CCA secure if for all probabilistic polynomial time adversaries $\mathcal{A}$, there is a negligible function $negl(n)$ such that $\mathrm{Adv}_{\Pi,\mathcal{A}}^{FHE-CCA}(n)\overset{def}{=}\left|\Pr\left[\mathrm{EXP}_{FHE,\Pi,\mathcal{A}}^{(b)}(n)=b^*\right]-\frac{1}{2}\right|\le negl(n)$, Where for each $b\in\{0,1\}$ and $n\in\mathbb{N}$ the experiment $\mathrm{EXP}_{FHE,\Pi,\mathcal{A}}^{(b)}(n)$ is defined as:

$$EXP_{FHE,\Pi,\mathcal{A}}^{(b)}(n)=[(PK, SK, EK)\leftarrow KeyGen(1^n);$$

$$(b_0^*, b_1^*, \mathrm{State})\leftarrow\mathcal{A}^{\mathcal{O}}(\mathrm{find}, PK); b^*\xleftarrow{\$}\{b_0^*, b_1^*\};$$

$$C^*\leftarrow Enc(PK, b^*); b\leftarrow\mathcal{A}^{\mathcal{O}}(\mathrm{guess}, \mathrm{State}, C^*); b=b^*]$$

The $\mathcal{O}$ consists of the three oracles **Dec**, **Eval** and **RevEK** defined as follows.

Setup. The challenger obtain a public key *PK*, a decryption key *SK* and an evaluation key *EK* by running $KeyGen(1^n)$. It gives the public key *PK* to the adversary $\mathcal{A}$. What's more, the challenger maintains a list $\mathcal{D}$ ($\mathcal{D}$ is set as $\varnothing$ initially).

Query1. The adversary $\mathcal{A}$ adaptively issues the following queries:

1. The evaluation oracle **RevEK**: The challenger sends the evaluation key *EK* to $\mathcal{A}$.
2. The decryption oracle **Dec**: The challenger uses the key *SK* to decode *C* with algorithm *Dec*. The result is sent back to $\mathcal{A}$. This oracle is not available if $\mathcal{A}$ has queried to RevEK.

3. The evaluation oracle **Eval**: The challenger runs algorithm *Eval* (*EK, f,* C_1, …, C_k) to obtain a ciphertext *C*, which is returned to $\mathcal{A}$. This oracle is not available if $\mathcal{A}$ has queried to RevEK.

Challenge. The challenge first selects a message bit $b^* \in \{0,1\}$ uniformly at random. Then, it computes $C^* \leftarrow Enc(PK, b^*)$, and sends the challenge ciphertext C^* to the adversary. Finally, the challenger updates the list by $\mathcal{D} \leftarrow \mathcal{D} \cup \{C^*\}$.

Query2. The adversary $\mathcal{A}$ adaptively issues the following queries:

1. The evaluation oracle **RevEK**: The challenger sends the evaluation key *EK* to $\mathcal{A}$.
2. The decryption oracle **Dec**: If $C \in \mathcal{D}$, the challenger returns $\perp$. Otherwise, it is the same as the Dec oracle in query 1.
3. The evaluation oracle **Eval**: It is the same as the Eval oracle in query 1. In addition, if there exists $i \in [k]$ such that $C_i \in \mathcal{D}$, then the challenger updates the list by $\mathcal{D} \leftarrow \mathcal{D} \cup \{C_i\}$.

Guess. The adversary $\mathcal{A}$ outputs its guess $b \in \{0,1\}$ for b^* and wins the game if $b = b^*$.

The advantage of the adversary in this game is defined as $\left| \Pr\left[b = b^* \right] - \frac{1}{2} \right|$ where the probability is taken over the random bits used by the challenger and the adversary.

3 Construction

3.1 Building Blocks

In this part, we introduce the multi-identity IBFHE and one-time strong signature. Multi-Identity IBFHE scheme allows to evaluate on ciphertexts created with different identity. Now we give the definition of multi-identity IBFHE and its IND-sID-CPA security.

Definition 8 (Multi-Identity IBFHE). Let $\mathcal{M}$ be a message space, $\mathcal{I}$ be an identity space, and $\mathbb{C}$ be a collection of circuits $f : \mathcal{M}^k \to \mathcal{M}$. A Multi-Identity (Leveled) IBFHE scheme is a 5-tuple of probabilistic polynomial time algorithms (IBFHE.Setup, IBFHE.KeyGen, IBFHE.Encrypt, IBFHE.Decrypt, IBFHE.Eval) defined as follows:

- IBFHE.Setup $(1^n, L)$ takes as input a security parameter *n* and a number of levels *L*(circuit depth to support) . It outputs a public parameters *pp*, a master secret key *msk* and an evaluation key *evk*.
- IBFHE.KeyGen (msk, id) takes as input the master secret key *msk* and an identity *id*. It outputs a secret key sk_{id} for identity *id*.
- IBFHE.Enc(pp, id, μ) takes as input public parameters *pp*, an identity *id* , and a message $\mu \in \mathcal{M}$. It outputs a ciphertext *CT* that encrypts μ under identity *id*.
- IBFHE.Dec$(sk_{id_1}, \cdots, sk_{id_d}, CT)$ takes as input *d* secret keys $sk_{id_1}, \cdots, sk_{id_d}$ for identities $id_1, \cdots, id_d$ and a ciphertext *CT*. It outputs $\mu' \in \mathcal{M}$ if *CT* is a valid encryption under identities $id_1, \cdots, id_d$, outputs a failure symbol $\perp$ otherwise.
- IBFHE.Eval $(evk, f, CT_1, \cdots, CT_k)$ takes as input the evaluation key *evk* , a circuit *f* and a tuple of ciphertexts $CT_1, \cdots, CT_k$. It outputs an evaluated ciphertext *CT*.

Informally, the scheme meets the decryption correctness and evaluation correctness. Over all choice of $(pp, msk, evk) \leftarrow$ IBFHE.Setup $(1^n, L)$, $id_1, \cdots, id_d \in \mathcal{I}$, $f : \mathcal{M}^k \to \mathcal{M} \in \{f \in \mathbb{C} : depth(f) \le L\}$, $\mu_1, \ldots, \mu_k \in \mathcal{M}$, $CT_i \leftarrow$ IBFHE.Enc(pp, id_j, μ_i) for $i \in [k]$ and ciphertext $CT' \leftarrow$ IBFHE.Eval $(evk, f, CT_1, \cdots, CT_k)$,satisfying:

IBFHE.Dec$(sk_{id_1}, \cdots, sk_{id_d}, CT') = f(\mu_1, \ldots, \mu_k)$ for $j \in [d]$

and $sk_{id_j} \leftarrow$ IBFHE.KeyGen (msk, id_j) .

The selective-identity IND-CPA security game for multi-identity IBFHE is the same as that for standard identity-based encryption. See [8] for detail. There are two multi-identity IBFHE schemes in the literature: the scheme of Clear and McGoldrick[9] and our related scheme which will be published in Journal of Cryptologic Research.

One-time strong signature A "strong" signature scheme has the property that it is infeasible to create new valid signature even for preciously-signed messages. A one-time strong signature scheme $\mathcal{S}$ consists of three algorithms $\mathcal{S} = (\text{Gen}, \text{Sign}, \text{Vrfy})$, $\mathcal{S}.\text{Gen}(1^k)$ generates a key-pair (vk, sk) , $\mathcal{S}.\text{Sign}(sk, m)$ outputs a signature σ of message *m*, and $\mathcal{S}.\text{Vrfy}(vk, \sigma)$ outputs 1 when σ is a valid signature of *m*, otherwise outputs $\perp$. We point out that this signature scheme be secure in the sense that an adversary is unable to forge even a new signature on a previously-signed message, which is called strong unforgeability.

3.2 Lattice and LWE

The main idea behind our approach is to exploit multi-identity IBFHE and one-time strong signature to construct a CCA-secure FHE scheme.

The public key of our proposed FHE scheme is the public parameters of the multi-identities IBFHE scheme, the secret key is the corresponding master key, and the evaluation key is (vk', sk', evk) , where (vk', sk') is a key-pair for the signature scheme and *evk* is generated by the multi-identity IBFHE scheme.

To encrypt a message bit, the encryption algorithm first runs algorithm $\mathcal{S}$.Gen to obtain a key-pair (*vk*, *sk*), and then uses the multi-identity IBFHE scheme to encrypt the message bit with respect to the "identity" *vk*, with the resulting ciphertext denoted as *CT*. Next, the signing key *sk* is used to sign *CT* to obtain a signature σ . The final ciphertext *C* consists of the verification key *vk*, the multi-identity IBFHE ciphertext *CT* and the signature σ . Given a ciphertext $C = (vk, CT, \sigma)$, the

decryption algorithm first uses algorithm $\mathcal{S}$.Vrfy to verify the signature σ on CT with respect to vk and output $\perp$ if the verification fails. Otherwise, the decryption algorithm generates the private key sk_{vk} corresponding to the "identity" vk, and decrypts the ciphertext CT using the underlying multi-identity IBFHE scheme.

Given a tuple of ciphertexts $C = (C_1,\ldots,C_k)$ where $C_i = (vk_i, CT_i, \sigma_i)$, and a Boolean circuit $f:\{0,1\}^k \to \{0,1\}$, the evaluation algorithm first verifies the signature σ_i on CT_i with respect to vk_i for each $i \in [k]$ and outputs $\perp$ if the verification fails. Otherwise, the evaluation algorithm evaluates the Boolean circuit f on the ciphertexts $CT_1,\ldots,CT_k$ using the underlying multi-identity IBFHE scheme. Then the resulting ciphertext CT' is signed using sk' to obtain a signature σ', and the evaluation algorithm outputs the ciphertext $C' = (CT', vk', \sigma')$.

3.3 Constructed CCA Secure FHE Scheme

Given a multi-identity IBFHE scheme IBFHE= (Setup, KeyGen, Encrypt, Decrypt, Eval) secure against selective-identity chosen-plaintext attacks, we construct a CCA-secure FHE scheme. In the construction, we use a one-time strong signature scheme $\mathcal{S}$= (Gen,Sign,Vrfy) in which the verification key vk output by Gen has length ℓ, and ℓ is the length of identities for multi-identity IBFHE scheme. We now present our construction of CCA-secure FHE scheme.

Setup On input a security parameter λ, and a number of levels L. The setup algorithm runs IBFHE.Setup$(1^\lambda, L)$ to obtain (pp, msk, evk), and runs $\mathcal{S}$.Gen(1^λ) to obtain a key-pair (vk', sk'). Output the public key PK=pp, the secret key SK=msk, and the evaluation key $EK = (vk', sk', evk)$.

Encryption On input the public key PK, and a message bit $\mu \in \{0,1\}$, the following steps are performed:

1. Run $\mathcal{S}$.Gen(1^λ) to obtain a key-pair (vk, sk).
2. Compute IBFHE.Enc$(pp, id = vk, \mu)$ to get a cipher CT, and $\mathcal{S}$.Sign(sk, CT) to get a sign σ.
3. Output the ciphertext $C = (CT, vk, \sigma)$.

Decryption To decrypt a ciphertext $C = (CT, vk, \sigma)$ using secret key SK. It proceeds as follows.

1. Check whether $\mathcal{S}$.Vrfy$(vk, CT, \sigma) = 1$. If not, it outputs $\perp$ and abort.
2. Compute IBFHE.KeyGen (msk, vk_i) to obtain sk_{vk_i} for all $i = 1,\cdots,k$.
3. Run Decrypt $(sk_{vk_1},\cdots,sk_{vk_k}, C)$ to obtain μ, and output the message bit μ.

Evaluation On input the public key PK=pp, the evaluation key $EK = (vk', sk', evk)$, a circuit $f \in \mathbb{C}$ and a tuple of ciphertexts $\boldsymbol{C} = C_1 = (CT_1, vk_1, \sigma_1),\cdots, C_k = (CT_k, vk_k, \sigma_k)$. It proceeds as follows.

1. Check whether $\mathcal{S}$.Vrfy$(vk_i, CT_i, \sigma_i) = 1$, for all $i = 1,\cdots,k$. If not, it outputs $\perp$ and abort.
2. Compute IBFHE.Eval $(evk, f, CT_1,\cdots,CT_k)$ to get CT'.
3. Run $\mathcal{S}$.Sign(sk', CT') to get a sign σ'.
4. Output the ciphertext $C = CT', (vk', vk_1,\cdots,vk_k), \sigma'$.

3.3 Correctness and Security

Correctness. If the underlying multi-identity IBFHE scheme satisfies encryption correctness and evaluation correctness, it is obvious that the above construction of FHE satisfies the correctness requirements.

Now, we give the CCA-secure proof of the proposed FHE scheme.

Theorem 1. If the underlying multi IBFHE scheme is IND-sID-CPA secure, and the signature scheme $\mathcal{S}$ is strongly EUF-CMA secure, then our proposed FHE scheme is CCA-secure.

Proof. To prove the CCA security of our proposed FHE scheme, we consider the following games which are described by its modification from the previous game.

Game0. This is the original CCA security game between an adversary $\mathcal{A}$ and a challenger $\mathcal{B}$ against our scheme.

Game1. Let C^* = (vk^*, CT^*, σ^*) be the challenge ciphertext, we slightly change the way that the challenger $\mathcal{B}$ answers the adversary's **Dec** and **Eval** queries. When the adversary $\mathcal{A}$ issues a **Dec** query on ciphertext C=(vk,CT,σ), the challenger $\mathcal{B}$ checks whether $vk = vk^*$, C=C^* and $\mathcal{S}$.Vrfy(vk,CT,σ) =1. If so, the challenger returns $\perp$; otherwise, it responds as in Game 0. When the adversary issues an **Eval** query on ciphers $(C_1,\cdots,C_k)$ and circuit f. For each $C_i = (vk_i, CT_i, \sigma_i)$, the challenger $\mathcal{B}$ checks whether there exists $i \in [k]$ such that $vk_i = vk^*$, $C_i \neq C^*$ and S.Vrfy$(vk_i, CT_i, \sigma_i) = 1$. If so, $\mathcal{B}$ returns $\perp$; otherwise, it responds as in Game 0.

To proof Game 0 and Game 1 are computationally indistinguishable, we define event E: $\mathcal{A}$ query on ciphertext $C = (vk, CT, \sigma)$ such that $vk = vk^*$, $C \neq C^*$, S.Vrfy$(vk, CT, \sigma) = 1$. If E does not happen, Game 0 is identical to Game 1. Meanwhile, if E happens with non-negligible probability, we can build an algorithm that breaks strong EUF-CMA security of the signature scheme $\mathcal{S}$ with non-negligible probability. So the Game 0 and Game 1 are computationally indistinguishable.

Game2. At the setup phase, except for the list $\mathcal{D}$, the challenger $\mathcal{B}$ also maintains another list $\mathcal{G}$, which is set as $\varnothing$ initially. We also modify the way how the adversary $\mathcal{A}$'s **Dec** and **Eval** queries are answered. Let $PK, SK, EK = (vk', sk', evk)$ be the public key, decryption key and evaluation key respectively.

When the adversary $\mathcal{A}$ issues a **Dec** query on ciphertext $C = (vk, CT, \sigma)$, the challenger checks whether $vk = vk^*$ or $vk \neq vk'$. If so, the challenger responds as in Game 1; otherwise, it proceeds as follows:

1. Check whether S.Vrfy$(vk, CT, \sigma) = 1$. If not, return $\perp$;

2. Search the list $\mathcal{G}$ for a record (μ, C). If such record does not exist, return $\perp$; otherwise, send μ to $\mathcal{A}$.

When the adversary $\mathcal{A}$ issues an **Eval** query on $(C_1, \cdots, C_k)$ and circuit *f*. $\mathcal{A}$ checks whether there exists $i \in [k]$ such that one of the following conditions holds: (1) $vk_i = vk^*, S.\text{Vrfy}(vk_i, CT_i, \sigma_i) = 1$ and $C_i \neq C^*$; (2) $vk_i = vk', S.\text{Vrfy}(vk_i, CT_i, \sigma_i) = 1$ and the list $\mathcal{G}$ does not contain a record (μ_i, C_i). If so, the challenger returns $\perp$ to $\mathcal{A}$; otherwise, the challenger runs $Eval(EK, SK, \boldsymbol{C}, f)$ to obtain a ciphertext *C*, which is returned to $\mathcal{A}$. In addition, when the ciphertext $C \neq \perp$, the challenger checks whether there exists $i \in [k]$ such that $C_i \in \mathcal{D}$. If so, the challenger updates the list by $\mathcal{D}=\mathcal{D} \cup \{C\}$; otherwise, it proceeds as follows.

1. For each $i \in [k]$, if $vk_i = vk'$, the challenger finds the record (μ_i, C_i) in the list $\mathcal{G}$; otherwise, the challenger uses the decryption key *SK* to decrypt C_i with algorithm Dec and obtain a message bit μ_i.
2. The challenger computes $\mu = f(\mu_1, \cdots, \mu_k)$ and updates the list by $\mathcal{G}=\mathcal{G} \cup \{(\mu, C)\}$.

Game 2 is the same as Game 1 except for the way of answering the adversary $\mathcal{A}$'s **Dec** and **Eval** queries when $\mathcal{A}$ submits a ciphertext $C = (vk, CT, \sigma)$ such that $vk = vk'$, $S.\text{Vrfy}(vk, CT, \sigma) = 1$. Recall that in our security definition of FHE, the adversary cannot issue the decryption or evaluation queries if it has requested the evaluation key. Since our proposed scheme satisfies the requirement of evaluation correctness, it is easy to observe that when $\mathcal{A}$ submits a ciphertext $C = (vk = vk', CT, \sigma)$ during its **Dec** or **Eval** queries where *C* is the return of $\mathcal{A}$'s some **Eval** query, the challenger's response is identical in Game 1 and Game 2. Define event E: the adversary $\mathcal{A}$ submits a ciphertext $C = (vk = vk', CT, \sigma)$ during its **Dec** or **Eval** queries such that $S.\text{Vrfy}(vk, CT, \sigma) = 1$ and *C* is not the response to $\mathcal{A}$'s some **Eval** query. If E does not happen, Game 1 is identical to Game 2. One can easily prove that if the signature scheme $\mathcal{S}$ is strongly EUF-CMA-secure, then event E happens with negligible probability.

Suppose there exist an adversary $\mathcal{A}$ that achieves a non-negligible advantage in Game 2. Then we can build an algorithm $\mathcal{B}$ that makes use of $\mathcal{A}$ to attack the underlying convertible IBFHE scheme in the IND-sID-CPA security game with a non-negligible advantage.

We prove these games are computationally indisting-uishable, and the advantage of the adversary is negligible in Game2. Therefore, we conclude that the advantage of the adversary in Game 0 (i.e., the original CCA security game) is negligible. This completes the proof of Theorem 1.

4 Conclusions

FHE encryption can effectively protect the user's privacy and data security in the cloud environment. In this paper, we present a concrete construction of CCA-secure FHE in the standard model, utilizing the IND-sID-CPA secure multi-identity IBFHE and strongly EUF-CMA secure signature. Compared with the existing schemes, the proposed scheme is improved in both efficiency and security.

Acknowledgements

This work was supported by Natural Science Foundation of Henan province(162300410332).

References

[1] Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of 41rd ACM Symposium on Theory of Computing (STOC2009), Bethesda, Maryland, USA, May 31-June 2,2009: 169-178.

[2] Emara, k., Hanaoka, G., Ohtake, G., et al.: Chosen ciphertext secure keyed-homomorphic public-key encryption. In: PKC 2013. LNCS, vol. 7778, pp. 32-50. Springer, Herdelberg (2013).

[3] Libert, B., Peters, T., Joye, M., Yung, M.: Non-malleability from malleability: simulation-sound quasi-adaptive NIZK proofs and CCA1-secure encryption from homomorphic signatures. In: ECURO-CRYPY 2014. LNCS 8441, pp.514-532. Springer, Herdelberg (2014).

[4] Junzuo Lai, Robert, H., Changshe Ma, et al,: CCA-secure Keyed-Fully Homomorphic Encryption. In: PKC2016, LNCS 9614, pp. 70-98. Springer, Herdelberg (2016).

[5] Garg, S., Gentry, C., Halevi, S. et al. Candidate indis-tinguishability obfuscation and functional encryption for all circuits. In: Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on. IEEE, 2013: 40-49.

[6] Canetti, R., Haleci, S., Katz, J.:Chosen-ciphertext security from identity-based encryption. In: EUROCRYPT 2004. LNCS 3027, pp. 207-222. Springer, Herdelberg(2004).

[7] Hu, B. C., Wong, D. S., Zhang, Z., Deng, X..: Certificateless signature: a new security model and an improved generic construction. In:*Designs, Codes and Cryptography*, *42*(2), 109-126. (2007).

[8] Agrawal S, Boneh D, Boyen X. Efficient lattice (H)IBE in the standard model. Proceedings of EUROCRYPT 2010, Riviera, France, 2010:553-572.

[9] Clear M, McGoldrick C. Multi-identity and multi-key leveled FHE from learning with errors[C].Proceedings ofCRYPTO2015, Santa Barbara, CA, USA, August 16-20, 2015:630-656.

[10] Shamir A. Identity-based cryptosystems and signature schemes. Advances in cryptology-CRYPTO 1984, Springer, Berlin, 1984; 47-53 .

[11] Wang FH, Liu ZH, Wang CX. Full secure identity-based encryption scheme with short public key size over lattices in the standard model. Proceedings of the International Journal of Computer Mathematics, 2016, 93(6):854-863.