CCA-Secure Keyed-Fully Homomorphic Encryption

Junzuo Lai[1,2], Robert H. Deng[3], Changshe Ma[4], Kouichi Sakurai[5] and Jian Weng[1]*

[1] Department of Computer Science, Jinan University, China
{laijunzuo,cryptjweng}@gmail.com
[2] The State Key Laboratory of Integrated Services Networks,
Xidian University, China
[3] School of Information Systems, Singapore Management University, Singapore
{robertdeng}@smu.edu.sg
[4] School of Computer, South China Normal University, China
{chsma}@163.com
[5] Department of Computer Science & Communication Engineering,
Kyushu University, Japan
{sakurai}@inf.kyushu-u.ac.jp

Abstract. To simultaneously achieve CCA security and homomorphic property for encryption, Emura et al. introduced a new cryptographic primitive named keyed-homomorphic encryption, in which homomorphic ciphertext manipulations can only be performed by someone holding a devoted evaluation key which, by itself, does not enable decryption. A keyed-homomorphic encryption scheme should provide CCA2 security when the evaluation key is unavailable to the adversary and remain CCA1-secure when the evaluation key is exposed. While existing keyed-homomorphic encryption schemes only allow simple computations on encrypted data, our goal is to construct CCA-secure *keyed-fully homomorphic encryption* (keyed-FHE) capable of evaluating any functions on encrypted data with an evaluation key.

In this paper, we first introduce a new primitive called convertible identity-based fully homomorphic encryption (IBFHE), which is an IBFHE with an additional transformation functionality, and define its security notions. Then, we present a generic construction of CCA-secure keyed-FHE from IND-sID-CPA-secure convertible IBFHE and strongly EUF-CMA-secure signature. Finally, we propose a concrete construction of IND-sID-CPA-secure convertible IBFHE, resulting in the first CCA-secure keyed-FHE scheme in the standard model.

Keywords: chosen ciphertext security, fully homomorphic encryption, convertible identity-based fully homomorphic encryption

* Corresponding author

1 Introduction

Today's information services are increasingly storing data across many servers shared with other data owners. An example of this is cloud computing which has the great potential of providing various services to the society at significantly reduced cost due to aggregated management of elastic resources. Since software systems are not guaranteed to be bug-free and hardware platforms are not under direct control of data owners in such distributed systems, security risks are abundant. To mitigate users' privacy concern about their data, a common solution is to outsource data in encrypted form so that it will remain private even if data servers are not trusted or compromised. To not nullify the benefits of cloud computing, however, we need homomorphic encryption schemes that allow meaningful computations on encrypted data. Recently, in a breakthrough effort, Gentry [28] constructed a *fully homomorphic encryption* (FHE) scheme enabling anyone to compute *arbitrary* functions on encrypted data. On the other hand, security against chosen-ciphertext attack (CCA) [24, 42, 47] is now a commonly accepted standard security notion for encryption, and unfortunately, it is well-known that CCA security and the homomorphic property cannot be achieved simultaneously.

The incompatibility of CCA security and homomorphicity cannot be reconciled under the assumption that everyone can "freely" perform homomorphic operations on ciphertexts. Very recently, Emura et al. [25] showed that in the setting where homomorphic operations are performed in a "controlled" fashion, CCA security and homomorphicity can be simultaneously achieved. They suggested a new primitive called keyed-homomorphic encryption [25], where homomorphic ciphertext manipulations are only possible to a party holding a devoted evaluation key EK which, by itself, does not enable decryption. A keyed-homomorphic encryption scheme should provide CCA2 security when the evaluation key is unavailable to the adversary and remain CCA1 secure when EK is exposed. Emura et al. [25] presented a number of keyed-homomorphic encryption schemes through hash proof systems [22], which only allow simple computations on encrypted data (i.e., either adding or multiplying encrypted ciphertexts, *but not both operations at the same time*). This paper is motivated by the goal of constructing CCA-secure *keyed-fully homomorphic encryption* (keyed-FHE)[1] capable of evaluating *any* functions on encrypted data with a devoted evaluation key EK.

Our Contribution. One may hope to obtain CCA-secure keyed-FHE by using the double encryption methodology: a ciphertext of an "inner" CPA-secure FHE scheme is encrypted by an "outer" CCA-secure encryption scheme, and the evaluation key EK is the decryption key of the "outer" CCA-secure encryption scheme. Unfortunately, this naive construction is not secure in the sense of our security definition for keyed-fully homomorphic encryption. An adversary is allowed to issue decryption queries before the evaluation key EK is exposed to

[1] We focus on *leveled* keyed-FHE schemes, and typically omit the term "leveled". In a *leveled* keyed-FHE scheme, the parameters of the scheme may depend on the *depth*, but not the *size*, of the circuits that the scheme can evaluate.

the adversary in our security definition. However, no such decryption query is allowed in the CPA security game of the underlying "inner" FHE scheme[2].

We propose a generic paradigm of constructing CCA-secure keyed-FHE, which follows the line of CHK transformation [18]. It is worth noting that, one cannot achieve CCA-secure keyed-FHE from IND-sID-CPA-secure IBFHE by CHK transformation directly, since each IBE ciphertext is under a *fresh* identity and the homomorphic evaluation functionality of IBFHE does not work.

- We define a new primitive named convertible identity-based fully homomorphic encryption (IBFHE) and its IND-sID-CPA security notions. Informally, a convertible IBFHE is an IBFHE with an additional transformation functionality, which may be of independent interest.
- Based on our new primitive, IND-sID-CPA-secure convertible IBFHE, and strongly EUF-CMA-secure signature, we propose a generic paradigm of constructing CCA-secure keyed-FHE by modifying CHK transformation [18] slightly.
- We construct a convertible identity-based (leveled) FHE scheme based on the adaptively-secure IBE scheme proposed by Agrawal et al. [1], and prove that it is IND-sID-CPA secure in the standard model, resulting in the first CCA-secure keyed-FHE scheme in the standard model. Actually, one can use our techniques to construct convertible IBFHE schemes based on the adaptively-secure IBE schemes proposed in [19, 2].

Convertible IBFHE. A convertible IBFHE scheme consists of seven algorithms: Setup, Extract, GenerateTK, Encrypt, Transform, Decrypt and Evaluate. Among these algorithms, (Setup, Extract, Encrypt, Decrypt, Evaluate) constitute the traditional IBFHE scheme; algorithms GenerateTK and Transform provide the following functionality: given a transformation key $\mathsf{TK}_{\mapsto \widetilde{\mathsf{ID}}}$ for an identity $\widetilde{\mathsf{ID}}$, which is generated by an authority using algorithm GenerateTK, one with the help of algorithm Transform can transform a ciphertext CT under *any* identity into a ciphertext under identity $\widetilde{\mathsf{ID}}$ without changing the underlying plaintext of CT.

The additional functionality of convertible IBFHE is reminiscent of identity-based proxy re-encryption (IBPRE) [32]. Unlike convertible IBFHE, in an IBPRE scheme, a transformation key (i.e., re-encryption key) $\mathsf{TK}_{\mathsf{ID}_1 \to \mathsf{ID}_2}$ associated with two identities ID_1 and ID_2, is generated by the user with identity ID_1, and one with the transformation key can *only* convert an encryption under identity ID_1 into the encryption under identity ID_2.

[2] Another naive approach to construct CCA-secure keyed-FHE is to utilize Naor-Yung paradigm [42]: a plaintext is encrypted twice (independently) by CPA-secure FHE, and then a non-malleable non-interactive zero-knowledge (NIZK) [51] proof is used in order to prove that both ciphertexts are encryptions to the same plaintext (the CRS needed for the NIZK is part of the public key); the evaluation key EK is the trapdoor associated with the CRS. However, as the construct by using the double encryption methodology, this construction is not secure in the sense of our security definition: the adversary is allowed to use the decryption oracle even after the challenge phase, just before the adversary requests EK.

The adaptive security of convertible IBFHE requires that given a challenge ciphertext CT^* under some identity ID^*, no PPT adversary can distinguish, except with a negligible advantage, whether CT^* is an encryption of 1 under identity ID^* or an encryption of 0 under identity ID^*. We allow an adversary to adaptively issue private key queries on identities ID and transformation key queries on identities $\widetilde{\mathsf{ID}}$, but with the natural constraints that: 1) the adversary cannot issue private key query for the challenge identity ID^*; 2) the adversary cannot issue private key query for an identity $\widetilde{\mathsf{ID}}$ such that the adversary has issued a transformation key query on $\widetilde{\mathsf{ID}}$, and *vice versa*.

For constructing CCA-secure keyed-FHE, we only require that the underlying convertible IBFHE be secure in a weaker security model, denoted as $\mathsf{IND\text{-}sID\text{-}CPA}$ security model. In this weaker security model, the transformation key query can be issued only *once* by the adversary, and the target identity ID^* and the designated identity $\widetilde{\mathsf{ID}}$ which the adversary wants to obtain the corresponding transformation key must be committed by the adversary ahead of the system setup.

CCA-secure keyed-FHE from IND-sID-CPA-secure convertible IBFHE. We give a high-level description on how to construct a CCA-secure keyed-FHE scheme from an $\mathsf{IND\text{-}sID\text{-}CPA}$-secure convertible IBFHE scheme characterized by $(\mathsf{GenerateTK}, \mathsf{Transform})$, with the help of a strongly $\mathsf{EUF\text{-}CMA}$-secure signature scheme $\mathcal{S} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$.

The public key of our proposed keyed-FHE scheme is the public parameters of the convertible IBFHE scheme, the secret key is the corresponding master key, and the evaluation key is $(\widetilde{vk}, \widetilde{sk}, \mathsf{TK}_{\mapsto\widetilde{vk}})$, where $(\widetilde{vk}, \widetilde{sk})$ is a key-pair for the signature scheme $\mathcal{S}$ and $\mathsf{TK}_{\mapsto\widetilde{vk}}$ which is generated by algorithm $\mathsf{GenerateTK}$ of the convertible IBFHE scheme is the transformation key for "identity" $\widetilde{vk}$.

To encrypt a message bit, the encryption algorithm first runs algorithm $\mathcal{S}.\mathsf{Gen}$ to obtain a key-pair (vk, sk), and then uses the convertible 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 convertible IBFHE ciphertext CT and the signature σ. Given a ciphertext $C = (vk, \mathsf{CT}, \sigma)$, the decryption algorithm first uses algorithm $\mathcal{S}.\mathsf{Vrfy}$ to verify the signature σ on CT with respect to vk and outputs $\perp$ 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 convertible IBFHE scheme.

Given a tuple of ciphertexts $\boldsymbol{C} = (C_1, \ldots, C_k)$ where $C_i = (vk_i, \mathsf{CT}_i, \sigma_i)$, 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, for each $i \in [k]$, with $\mathsf{TK}_{\mapsto\widetilde{vk}}$, it runs algorithm $\mathsf{Transform}$ of the convertible IBFHE scheme to convert the ciphertext CT_i under "identity" vk_i into a ciphertext $\widetilde{\mathsf{CT}}_i$ under the "identity" $\widetilde{vk}$. Since now $\widetilde{\mathsf{CT}}_1, \ldots, \widetilde{\mathsf{CT}}_k$ are the ciphertexts under the same "identity" $\widetilde{vk}$, the evaluation algorithm can evaluate the Boolean circuit f on the ciphertexts $\widetilde{\mathsf{CT}}_1, \ldots, \widetilde{\mathsf{CT}}_k$ using the underlying convertible IBFHE scheme. Then the resulting ciphertext $\widetilde{\mathsf{CT}}$ is signed using $\widetilde{sk}$ to obtain a signature $\tilde{\sigma}$, and the evaluation algorithm outputs the ciphertext $C = (\widetilde{vk}, \widetilde{\mathsf{CT}}, \tilde{\sigma})$.

As for the security of our proposed keyed-FHE scheme, we show that if there exists an adversary $\mathcal{A}$ with a non-negligible advantage in the CCA security game, we can create a reduction algorithm $\mathcal{B}$ that breaks the $\mathsf{IND\text{-}sID\text{-}CPA}$ security of the underlying convertible IBFHE scheme. The reduction algorithm $\mathcal{B}$ is informally described as follows. $\mathcal{B}$ first runs $\mathcal{S}.\mathsf{Gen}$ to obtain two key-pairs (vk^*, sk^*) and $(\widetilde{vk}, \widetilde{sk})$. Then, $\mathcal{B}$ sets vk^* and $\widetilde{vk}$ as its target "identity" and designated "identity", which are submitted to its challenger in the $\mathsf{IND\text{-}sID\text{-}CPA}$ security game of the convertible IBFHE scheme. $\mathcal{B}$ is given the public parameters of the convertible IBFHE scheme and the transformation key $\mathsf{TK}_{\mapsto \widetilde{vk}}$ for "identity" $\widetilde{vk}$. Now, $\mathcal{B}$ can use $(\widetilde{vk}, \widetilde{sk})$ and $\mathsf{TK}_{\mapsto \widetilde{vk}}$ to answer $\mathcal{A}$'s evaluation queries and the evaluation key query, and the challenge ciphertext C^* given to $\mathcal{A}$ is set as $(vk^*, \mathsf{CT}^*, \sigma^*)$, where CT^* is $\mathcal{B}$'s challenge ciphertext of the convertible IBFHE scheme and $\sigma^* \leftarrow \mathcal{S}.\mathsf{Sign}(sk^*, \mathsf{CT}^*)$. Next, we shall explain how $\mathcal{B}$ answers the decryption queries for ciphertexts $C = (vk, \mathsf{CT}, \sigma)$ issued by adversary $\mathcal{A}$.

We say a ciphertext $C = (vk, \mathsf{CT}, \sigma)$ is valid if σ is a valid signature on CT with respect to vk. For $\mathcal{A}$'s decryption query on a ciphertext $C = (vk, \mathsf{CT}, \sigma)$ such that C is a valid ciphertext and $vk \notin \{vk^*, \widetilde{vk}\}$, $\mathcal{B}$ can issue a private key query on the "identity" vk to its challenger to obtain the corresponding private key SK_{vk}, and use the private key SK_{vk} to answer $\mathcal{A}$'s query. The subtlety lies in how $\mathcal{B}$ deals with $\mathcal{A}$'s decryption query on a valid ciphertext $C = (vk, \mathsf{CT}, \sigma)$ such that $vk \in \{vk^*, \widetilde{vk}\}$. Recall that $\mathcal{B}$ is not allowed to issue a private key query on the "identity" $vk \in \{vk^*, \widetilde{vk}\}$ to it's own challenger in the $\mathsf{IND\text{-}sID\text{-}CPA}$ security game of the convertible IBFHE scheme. We first note that any valid ciphertext $C = (vk, \mathsf{CT}, \sigma)$ submitted by the adversary during its queries must, except with negligible probability, have $vk \neq vk^*$ by the strong security of the signature scheme $\mathcal{S}$. The crux of the security proof is then to show how $\mathcal{B}$ answers $\mathcal{A}$'s decryption query on a valid ciphertext $C = (vk, \mathsf{CT}, \sigma)$ such that $vk = \widetilde{vk}$.

In our security definition of keyed-fully homomorphic encryption, the adversary can issue the decryption and evaluation queries only if it does not request the evaluation key to be exposed. Hence, for any valid ciphertext $C = (vk, \mathsf{CT}, \sigma)$ submitted by the adversary during its decryption queries, if $vk = \widetilde{vk}$, with overwhelming probability, C is one of $\mathcal{B}$'s responses to $\mathcal{A}$'s evaluation queries by the strong $\mathsf{EUF\text{-}CMA}$ security of the signature scheme $\mathcal{S}$. Based on the above observation, $\mathcal{B}$ will resort to a list EList to answer $\mathcal{A}$'s decryption query on a valid ciphertext $C = (vk = \widetilde{vk}, \mathsf{CT}, \sigma)$. The list EList is set as $\emptyset$ initially and is updated while answering $\mathcal{A}$'s evaluations queries. Now, when $\mathcal{A}$ issues an evaluation query on a tuple of ciphertext $\boldsymbol{C} = (C_1, \ldots, C_k)$ and a Boolean circuit f, after sending the result C of the evaluation algorithm to the adversary, $\mathcal{B}$ additionally proceeds as follows.

1. Check whether there exists an $i \in [k]$ such that $C_i = C^*$. If so, update the list by $\mathsf{EList} \leftarrow \mathsf{Elist} \cup \{(\perp, C)\}$. Note that, to avoid an unachievable security definition, $\mathcal{B}$ answers $\perp$ for "unallowable ciphertext" that are the result of homomorphic evaluation for C^* and any ciphertext of $\mathcal{A}$'s choice.
2. For each valid ciphertext $C_i = (vk_i, \mathsf{CT}_i, \sigma_i)$ where $i \in [k]$, obtain the corresponding plaintext b_i by finding the corresponding record (b_i, C_i) in the list EList if $vk_i = \widetilde{vk}$ or decrypting CT_i with the help of issuing a private key query on the "identity" vk_i to its challenger. Then, compute the message bit $m = f(b_1, \ldots, b_k)$ and update the list by $\mathsf{EList} \leftarrow \mathsf{Elist} \cup \{(m, C)\}$.

Consequently, when $\mathcal{A}$ issues a decryption query on a valid ciphertext $C = (vk, \mathsf{CT}, \sigma)$ such that $vk = \widetilde{vk}$, except with negligible probability, $\mathcal{B}$ can find a record (m, C) in the list EList and return m to the adversary as its answer. Hence, by the strong $\mathsf{EUF\text{-}CMA}$ security of the signature scheme $\mathcal{S}$, with overwhelming probability, $\mathcal{B}$ simulates the CCA security game of our proposed keyed-FHE scheme for $\mathcal{A}$ properly. Therefore, if $\mathcal{A}$ has a non-negligible advantage in the CCA security game, $\mathcal{B}$ breaks the $\mathsf{IND\text{-}sID\text{-}CPA}$ security of the underlying convertible IBFHE scheme with a non-negligible advantage.

Construction of IND-sID-CPA-secure convertible IBFHE. Based on the standard learning with errors (LWE) problem [49], Agrawal et al. [1] proposed an efficient identity-based encryption scheme and showed that their base construction can be extended to an adaptively-secure IBE using a lattice analog of the Waters IBE [56]. Our $\mathsf{IND\text{-}sID\text{-}CPA}$-secure convertible IBFHE starts from the adaptively-secure IBE scheme in [1].

An encryption of a message bit b for an identity $\mathsf{ID} = (d_1, \ldots, d_\ell) \in \{-1, 1\}^\ell$ in the adaptively-secure IBE scheme [1] takes the form of

$$c_0 = u^\top s + x + b\lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q, \ c_1 = F_{\mathsf{ID}}^\top s + \begin{bmatrix} y \\ R_{\mathsf{ID}}^\top y \end{bmatrix} \in \mathbb{Z}_q^{2m},$$

where $F_{\mathsf{ID}} = A \mid B_0 + \sum_{i=1}^{\ell} d_i B_i$, $R_{\mathsf{ID}} = \sum_{i=1}^{\ell} d_i R_i$, and $A, B_0, B_1, \ldots, B_\ell, u$ are the system's public parameters, a short basis T_A for $\Lambda_q^\perp(A)$ is the master key, $s, x, y, R_1, \ldots, R_\ell$ are noise vectors with short norm used in the encryption algorithm. The private key $\mathsf{SK}_{\mathsf{ID}}$ for identity ID is a short vector e_{ID} in $\Lambda_q^u(F_{\mathsf{ID}})$, hence the message bit b can be recovered from $c_0 - e_{\mathsf{ID}}^\top c_1$.

Agrawal et al. [1] utilized the partitioning strategy to prove the adaptively-secure security of the above IBE scheme. In the security reduction, $B_1, \ldots, B_\ell$ in the public parameters are set as $B_i = AR_i^* + h_i B_0$, where all the matrices R_i^* are random and h_i is a secret coefficient in $\mathbb{Z}_q$. Consequently,

$$F_{\mathsf{ID}} = A \mid B_0 + \sum_{i=1}^{\ell} d_i B_i = A \mid A(\sum_{i=1}^{\ell} d_i R_i^*) + h_{\mathsf{ID}} B_0,$$

where $h_{\mathsf{ID}} = (1 + \sum_{i=1}^{\ell} d_i h_i)$, and the identity space is partitioned into two parts according to whether h_{ID} is equal to 0 or not. If $h_{\mathsf{ID}} \neq 0$, the simulator, without knowing the master key, can use a trapdoor T_{B_0} for $\Lambda_q^{\perp}(B_0)$ to generate the private key for identity ID, i.e., a short vector e_{ID} in $\Lambda_q^{u}(F_{\mathsf{ID}})$. The simulator cannot produce the corresponding private key for identities ID such that $h_{\mathsf{ID}} = 0$, but will be able to construct a useful challenge to solve the given LWE problem instance. Let ID^* be the challenge identity and let $\mathsf{ID}_1, \ldots, \mathsf{ID}_Q$ be the identities for which the adversary issues private key queries. The security proof will require that for any $\mathsf{ID}^*, \mathsf{ID}_1, \ldots, \mathsf{ID}_Q$, with non-negligible probability,

$$h_{\mathsf{ID}^*} = 0 \wedge h_{\mathsf{ID}_1} \neq 0 \wedge \ldots \wedge h_{\mathsf{ID}_\ell} \neq 0,$$

which can be satisfied by the abort-resistant hash family used in [56, 34, 7].

The idea of constructing convertible IBFHE is summarized as follows. We first show how to design a convertible IBE scheme (i.e. without the homomorphic evaluation functionality), and then extend it to a convertible IBFHE scheme. To construct convertible IBE, we should provide an approach to converting a ciphertext CT under *any* identity ID into a ciphertext $\widetilde{\mathsf{CT}}$ under the designated identity $\widetilde{\mathsf{ID}}$. For transformation correctness (i.e., decrypting CT and $\widetilde{\mathsf{CT}}$ with the corresponding private key $\mathsf{SK}_{\mathsf{ID}}$ for identity ID and $\mathsf{SK}_{\widetilde{\mathsf{ID}}}$ for identity $\widetilde{\mathsf{ID}}$ respectively, must have the same result), we need be able to check whether a ciphertext is *well-formed*. However, starting from the adaptively-secure IBE scheme proposed in [1], we are thrown into a dilemma. Agrawal et al. [1] proved that in their proposed IBE scheme, encryption of any message bit is indistinguishable from uniform vector over $\mathbb{Z}_q$ under the LWE assumption. That is, any well-formed ciphertext in [1] is pseudorandom; thus it is difficult to design a mechanism to check the well-formedness of a ciphertext. We resort to the recent advances in indistinguishability obfuscation [52] to overcome the obstacle.

Besides $A, B_0, B_1, \ldots, B_\ell, u$, the public parameters of our proposed convertible IBE include an indistinguishability obfuscation of the following program that takes as input an identity $\mathsf{ID} = (d_1, \ldots, d_\ell) \in \{-1, 1\}^\ell$, a message bit $b \in \{0, 1\}$ and randomness r,

1. Set $t = \mathsf{PRG}(r)$ and $(s, x, y, R_1, \ldots, R_\ell) = \mathsf{F}(\mathrm{K}, \mathsf{ID}, t)$;
2. Compute $c_0 = u^\top s + x + b\lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q$, $c_1 = F_{\mathsf{ID}}^\top s + \begin{bmatrix} y \\ R_{\mathsf{ID}}^\top y \end{bmatrix} \in \mathbb{Z}_q^{2m}$ and output (t, c_0, c_1).

The system's master key additionally includes the key K to the puncturable pseudorandom function (PRF) [52] F. Let $\mathbf{P}^{\mathsf{Enc}}$ be the above obfuscated program. The encryption algorithm simply runs $\mathbf{P}^{\mathsf{Enc}}(\mathsf{ID}, b, r)$ and outputs the result (t, c_0, c_1). The private key $\mathsf{SK}_{\mathsf{ID}}$ for identity ID is a short vector e_{ID} in $\Lambda_q^{u}(F_{\mathsf{ID}})$, and given a ciphertext (t, c_0, c_1) under identity ID, the message bit b can be recovered from $c_0 - e_{\mathsf{ID}}^\top c_1$. Observe that, with the knowledge of K, one can retrieve the randomness $s, x, y, R_1, \ldots, R_\ell$ and the message bit b from a ciphertext (t, c_0, c_1) under an identity ID, and thus can check the well-formedness of the ciphertext and re-encrypt b under another identity. Consequently, the transformation key for an designated identity $\widetilde{\mathsf{ID}} = (\tilde{d}_1, \ldots, \tilde{d}_\ell) \in \{-1, 1\}^\ell$ is an indistinguishability obfuscation of the following program that takes as input a ciphertext $\mathsf{CT} = (t, c_0, c_1)$ under an identity $\mathsf{ID} = (d_1, \ldots, d_\ell) \in \{-1, 1\}^\ell$,

1. Set $(s, x, y, R_1, \ldots, R_\ell) = \mathsf{F}(\mathrm{K}, \mathsf{ID}, t)$, and check whether there exists $b \in \{0,1\}$ such that $c_0 = u^\top s + x + b\lfloor \frac{q}{2} \rfloor$ and $c_1 = F_{\mathsf{ID}}^\top s + \begin{bmatrix} y \\ R_{\mathsf{ID}}^\top y \end{bmatrix}$. If not, output $\perp$.
2. Set $(\tilde{s}, \tilde{x}, \tilde{y}, \tilde{R}_1, \ldots, \tilde{R}_\ell) = \mathsf{F}(\mathrm{K}, \widetilde{\mathsf{ID}}, t)$, and compute $\tilde{c}_0 = u^\top \tilde{s} + \tilde{x} + b\lfloor \frac{q}{2} \rfloor$, $\tilde{c}_1 = F_{\widetilde{\mathsf{ID}}}^\top \tilde{s} + \begin{bmatrix} \tilde{y} \\ \tilde{R}_{\widetilde{\mathsf{ID}}}^\top \tilde{y} \end{bmatrix}$ and output $(t, \tilde{c}_0, \tilde{c}_1)$.

Let $\mathbf{P}^{\mathsf{Trans}}$ be the above obfuscated program. To convert an encryption CT under identity ID into the encryption under identity $\widetilde{\mathsf{ID}}$, the transformation algorithm now simply runs $\mathbf{P}^{\mathsf{Trans}}(\mathsf{ID}, \mathsf{CT})$ and outputs the result.

As for the $\mathsf{IND\text{-}sID\text{-}CPA}$ security of the convertible IBE scheme, we follow the line of [1], i.e., utilizing the partitioning strategy. Let ID^* be the challenge identity, $\widetilde{\mathsf{ID}}$ be the designated identity which the adversary wants to obtain the corresponding transformation key $\mathsf{TK}_{\mapsto \widetilde{\mathsf{ID}}} = (\widetilde{\mathsf{ID}}, \mathbf{P}^{\mathsf{Trans}})$, and $\mathsf{ID}_1, \ldots, \mathsf{ID}_Q$ be the identities for which the adversary issues private key queries. Let $\mathsf{CT}^* = (t^*, c_0^*, c_1^*)$ be the challenge ciphertext for ID^*. In the security reduction, there exist some subtleties:

1. It requires that $h_{\mathsf{ID}^*} = 0$, in order to construct the challenge $\mathsf{CT}^* = (t^*, c_0^*, c_1^*)$ to solve the given LWE problem instance. Like the security reduction in [1], the randomness s^*, x^*, y^* that are used to evaluate c_0^* and c_1^*, come from the given LWE problem instance and is unknown to the simulator. Hence, when the adversary runs $\mathbf{P}^{\mathsf{Trans}}(\mathsf{ID}^*, \mathsf{CT}^*)$, it will get an error symbol $\perp$, which enables it to distinguish the simulated settings and the real settings. We observe that the simulator can prepare CT^* and $\widetilde{\mathsf{CT}}^*$ at the setup phase, where $\widetilde{\mathsf{CT}}^*$ denotes the corresponding result of calling the transformation algorithm on the challenge ciphertext $\mathsf{CT}^* = (t^*, c_0^*, c_1^*)$, since in the $\mathsf{IND\text{-}sID\text{-}CPA}$ security game the adversary must commit ID^* and $\widetilde{\mathsf{ID}}$ ahead of the system setup. Consequently, the simulator can employ the technique of *punctured programs*, introduced by Sahai et al. [52], to simulate the transformation key for $\widetilde{\mathsf{ID}}$ properly.
2. It requires that for *any* $\mathsf{ID}^*, \widetilde{\mathsf{ID}}, \mathsf{ID}_1, \ldots, \mathsf{ID}_Q$, with non-negligible probability, $h_{\mathsf{ID}^*} = 0 \wedge h_{\widetilde{\mathsf{ID}}} = 0 \wedge h_{\mathsf{ID}_1} \neq 0 \wedge \ldots \wedge h_{\mathsf{ID}_\ell} \neq 0$. Unfortunately, it cannot be satisfied by the abort-resistant hash family used in [56, 34, 7]. On the other hand, we observe that, if ID^* and $\widetilde{\mathsf{ID}}$ are chosen uniformly at random, with non-negligible probability, the requirement of $h_{\mathsf{ID}^*} = 0 \wedge h_{\widetilde{\mathsf{ID}}} = 0 \wedge h_{\mathsf{ID}_1} \neq 0 \wedge \ldots \wedge h_{\mathsf{ID}_\ell} \neq 0$ can be satisfied by the abort-resistant hash family used in [56, 34, 7]. Therefore, we use another puncturable PRF to map an identity ID into a random identity $\mathsf{id} \in \{-1, 1\}^\ell$, and replace ID with id in all functionalities. Similarly, since ID^* and $\widetilde{\mathsf{ID}}$ must be committed by the adversary ahead of the system setup, the technique of *punctured programs* allows the simulator's simulation be performed properly.

So far, we obtain an $\mathsf{IND\text{-}sID\text{-}CPA}$-secure convertible IBE scheme. Next, we show that the convertible IBE scheme extends to a convertible IBFHE scheme. Recently, Gentry et al. [31] described a simple "compiler" that transforms any LWE-based IBE scheme (that satisfies certain natural properties) into an identity-based (leveled) FHE scheme. Since our proposed convertible IBE scheme starts from the LWE-based IBE schemes proposed in [1] that have the required properties, we can utilize the "compiler" to transform it into a convertible identity-based (leveled) FHE scheme.

Related Work. Emura et al. [25] showed that CCA security does not rule out homomorphicity when the capability to compute on encrypted data is controlled, by introducing a primitive called keyed-homomorphic encryption. Other approaches to reconcile homomorphism and non-malleability were taken in [44–46, 9, 20] but they inevitably satisfy weaker security notions than CCA security.

Based on hash proof systems [22], Emura et al. [25] constructed a number of CCA-secure keyed-homomorphic schemes. Recently, Libert et al. [40] applied linearly homomorphic structure-preserving signatures [39] to quasi-adaptive non-interactive zero-knowledge (QA-NIZK) proofs [37], proposed QA-NIZK proofs with unbounded simulation-soundness (USS), and constructed a CCA-secure keyed-homomorphic scheme with threshold decryption by applying USS. These CCA-secure keyed-homomorphic schemes only allow simple computations on encrypted data, i.e., either adding or multiplying encrypted ciphertexts, *but not both operations at the same time.*

Fully Homomorphic Encryption. The notion of fully homomorphic encryption (FHE) capable of performing any computations on encrypted data, was first put forward by Rivest et al. [50]. However, only in the past few years have candidate FHE schemes been proposed. The first such scheme was constructed by Gentry [28]; his work inspired a tremendous amount of research effort on improving the efficiency of his scheme [54, 53, 30, 29, 13, 21], realizations of FHE based on different assumptions [55, 16, 17, 14], and so on. Until now, fully homomorphic encryption schemes can only be proven secure against chosen-plaintext attack (CPA).

Controlled Homomorphic Encryption. Desmedt et al. [23] put forth the notion of a controllable homomorphic encryption scheme (CHES) that blends together the notion of a fully homomorphic encryption scheme and of a functional encryption scheme [8]. In a CHES, a *designated* homomorphic operation C can be efficiently performed on a *single* ciphertext by a party that has a special token for function C that is released by the owner of the secret key. Compared with CHES, keyed-FHE enables a party holding a devoted evaluation key to compute *arbitrary* functions on *ciphertexts*, and it can provide CCA2 security when the evaluation key is unavailable.

Indistinguishability Obfuscation. Program obfuscation deals with the problem of how to protect a program from reverse engineering while preserving functionality. Unfortunately, Barak et al. [6, 5] showed that the most natural simulation-based formulation of program obfuscation (a.k.a. "black-box obfuscation") is impossible to achieve for *general* programs in a very strong sense. Faced with this impossibility result, Barak et al. [6, 5] suggested another notion of program obfuscation named *indistinguishability obfuscation.* Roughly speaking, an indistinguishability obfuscation scheme ensures that the obfuscations of any two functionally equivalent circuits are computationally indistinguishable. Recently, Garg et al. [27] proposed the first candidate construction of an efficient indistinguishability obfuscation ($i\mathcal{O}$) for *general* programs.

Recently, staring with [52] there has been much interest in investigating what can be built from $i\mathcal{O}$, since this model leads to poly-time obfuscation of unrestricted program classes, circumventing the known impossibility results of [6, 5]. Subsequently, many papers [52, 48, 36, 57, 33, 35, 26, 11] have shown a wide range of cryptographic applications of $i\mathcal{O}$. We utilize $i\mathcal{O}$ to construct an IND-sID-CPA-secure convertible IBFHE scheme.

Organization. The rest of the paper is organized as follows. Some preliminaries are given in Section 2. We introduce the notion and security model of convertible IBFHE in Section 3. We propose a paradigm of constructing CCA-secure keyed-FHE from IND-sID-CPA-secure convertible IBFHE and strongly EUF-CMA-secure signature in Section 4. We present a concrete construction of IND-sID-CPA-secure convertible identity-based (leveled) FHE in Section 5. Section 6 concludes the paper.

2 Preliminaries

If S is a set, then $s_1, \dots, s_t \leftarrow S$ denotes the operation of picking elements $s_1, \dots, s_t$ uniformly at random from S. If $n \in \mathbb{N}$ then $[n]$ denotes the set $\{1, \dots, n\}$. For a probabilistic algorithm A, we denote $y \leftarrow A(x; R)$ the process of running A on input x and with randomness R, and assigning y the result. Let $\mathcal{R}_A$ denote the randomness space of A, and we write $y \leftarrow A(x)$ for $y \leftarrow A(x; R)$ with R chosen from $\mathcal{R}_A$ uniformly at random. A function $f(\kappa)$ is *negligible*, if for every $c > 0$ there exists a κ_c such that $f(\kappa) < 1/\kappa^c$ for all $\kappa > \kappa_c$. For a real $x \in \mathbb{R}$, $\lfloor x \rceil$ denotes the nearest integer to x, and $\lfloor x \rfloor$, $\lceil x \rceil$ for $x \geq 0$ to indicate rounding down or up.

2.1 Lattices

A full-rank lattice Λ is the set of all integer linear combinations of n linearly independent basis vectors belonging to some $\mathbb{R}^n$. In this work, we are interested in full-rank integer lattices that are restricted to $\mathbb{Z}^n$.

Definition 1 *Fixing q and given a matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$, define the following m-dimensional Ajtai lattices,*

$$\Lambda_q(\boldsymbol{A}) = \{\boldsymbol{y} \in \mathbb{Z}^m : \boldsymbol{y} = \boldsymbol{A}^T \boldsymbol{s} \bmod q \text{ for some } \boldsymbol{s} \in \mathbb{Z}^n\},$$
$$\Lambda_q^\perp(\boldsymbol{A}) = \{\boldsymbol{y} \in \mathbb{Z}^m : \boldsymbol{A}\boldsymbol{y} = \boldsymbol{0} \bmod q\}.$$

For any $\boldsymbol{u} \in \mathbb{Z}_q^n$ admitting an integral solution to $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{u} \bmod q$, define the coset (or shifted lattice) $\Lambda_q^{\boldsymbol{u}}(\boldsymbol{A}) = \{\boldsymbol{y} \in \mathbb{Z}^m : \boldsymbol{A}\boldsymbol{y} = \boldsymbol{u} \bmod q\} = \Lambda_q^\perp(\boldsymbol{A}) + \boldsymbol{x}$.

For a set of vectors $\boldsymbol{B} = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m\} \in \mathbb{Z}_q^{n\times m}$, denote by $\|\boldsymbol{B}\|$ the L_2 length of the longest vector in $\boldsymbol{B}$ and denote by $\widetilde{\boldsymbol{B}} = \{\widetilde{\boldsymbol{b}}_1, \ldots, \widetilde{\boldsymbol{b}}_m\}$ the Gram-Schmidt orthogonalization of $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m$ taken in that order. We refer to $\|\widetilde{\boldsymbol{B}}\|$ as the Gram-Schmidt norm of $\boldsymbol{B}$.

2.2 Discrete Gaussians

Let $\sigma \in \mathbb{R}^+$ and $\boldsymbol{c} \in \mathbb{R}^m$, the Gaussian function On $\mathbb{R}^m$ with center $\mathbf{c}$ and parameter σ is defined as $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \exp(-\pi\|\mathbf{x}-\mathbf{c}\|^2/\sigma^2)$. For a positive integer $m \in \mathbb{N}$, and a lattice $\Lambda \in \mathbb{Z}^m$, define the infinite discrete sum of Gaussian function over the lattice Λ, $\rho_{\sigma,\mathbf{c}}(\Lambda) = \sum_{\mathbf{x}\in\Lambda} \rho_{\sigma,\mathbf{c}}(\mathbf{x})$. The discrete Gaussian distribution $\mathcal{D}_{\Lambda,\sigma,\mathbf{c}}$ is the m-dimensional Gaussian distribution centered at $\mathbf{c}$ and restricted to the lattice Λ, defined as $\mathcal{D}_{\Lambda,\sigma,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{x})}{\rho_{\sigma,\mathbf{c}}(\Lambda)}$ for all the lattice point $\mathbf{x} \in \Lambda$. For ease of notation, we omit the center $\mathbf{c}$ if $\mathbf{c} = \mathbf{0}$, and then abbreviate $\mathcal{D}_{\Lambda,\sigma,\mathbf{0}}$ as $\mathcal{D}_{\Lambda,\sigma}$.

2.3 Sampling Algorithms

How to generate a random matrix $\boldsymbol{A}$ statistically close to uniform in $\mathbb{Z}_q^{n\times m}$ along with a short basis (i.e., trapdoor) $\boldsymbol{T}$ of $\Lambda_q^{\perp}(\boldsymbol{A})$ is an important technique in lattice-based cryptography. It has been widely investigated by [3, 4, 41]. We use the trapdoor sampling algorithm proposed by Alwen and Peikert [4].

Theorem 1 *Let $n \geq 1$ and q be an odd prime, and let $m \geq 6n\log q$. There is an efficient probabilistic polynomial-time algorithm* $\mathsf{TrapGen}(q, n)$ *that outputs $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ and $\boldsymbol{T} \in \mathbb{Z}^{m\times m}$ such that the distribution of $\boldsymbol{A}$ is within negl(n) statistical distance of uniform and $\boldsymbol{T}$ is a basis of $\Lambda_q^{\perp}(\boldsymbol{A})$ satisfying $\|\boldsymbol{T}\| \leq O(n\log q)$ and $\|\widetilde{\boldsymbol{T}}\| \leq O(\sqrt{n\log q})$ with all but negligible probability in n.*

In the construction and the simulation of our convertible IBFHE scheme, we employ the sampling algorithms $\mathsf{SampleLeft}$ and $\mathsf{SampleRight}$ given in [1], which can be used to sample relatively short vectors.

Theorem 2 *Let $\boldsymbol{A}$ be a rank n matrix in $\mathbb{Z}_q^{n\times m}$ and let $\boldsymbol{T_A}$ be a "short" basis of $\Lambda_q^{\perp}(\boldsymbol{A})$. Let $\boldsymbol{M}_1$ be a matrix in $\mathbb{Z}_q^{n\times m_1}$ and let $\boldsymbol{F}_1 = \boldsymbol{A}|\boldsymbol{M}_1$. Let $\boldsymbol{u}$ be a vector in $\mathbb{Z}_q^n$ and $\sigma > \|\widetilde{\boldsymbol{T_A}}\| \cdot \omega(\sqrt{\log(m+m_1)})$. There is a probabilistic polynomial-time algorithm* $\mathsf{SampleLeft}(\boldsymbol{A}, \boldsymbol{M}_1, \boldsymbol{T_A}, \boldsymbol{u}, \sigma)$ *that outputs a vector $\boldsymbol{e} \in \mathbb{Z}_q^{m+m_1}$ sampled from a distribution statistically close to $\mathcal{D}_{\Lambda_q^{\boldsymbol{u}}(\boldsymbol{F}_1),\sigma}$. In particular, $\boldsymbol{e} \in \Lambda_q^{\boldsymbol{u}}(\mathbf{F}_1)$.*

Theorem 3 *Let $\boldsymbol{B}$ be a rank n matrix in $\mathbb{Z}_q^{n\times m}$ and let $\boldsymbol{T_B}$ be a "short" basis of $\Lambda_q^{\perp}(\boldsymbol{B})$. Let $\boldsymbol{R}$ be a matrix in $\mathbb{Z}_q^{k\times m}$. Let $\boldsymbol{A}$ be a matrix in $\mathbb{Z}_q^{n\times k}$ and let $\boldsymbol{F}_2 = \boldsymbol{A}|\boldsymbol{A}\boldsymbol{R}+\boldsymbol{B}$. Let $\boldsymbol{u}$ be a vector in $\mathbb{Z}_q^n$ and $\sigma > \|\widetilde{\boldsymbol{T_B}}\| \cdot \|\boldsymbol{R}\| \cdot \omega(\sqrt{\log(m)})$. There is a probabilistic polynomial-time algorithm* $\mathsf{SampleRight}(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{R}, \boldsymbol{T_B}, \boldsymbol{u}, \sigma)$ *that outputs a vector $\boldsymbol{e} \in \mathbb{Z}_q^{m+k}$ sampled from a distribution statistically close to $\mathcal{D}_{\Lambda_q^{\boldsymbol{u}}(\boldsymbol{F}_1),\sigma}$. In particular, $\boldsymbol{e} \in \Lambda_q^{\boldsymbol{u}}(\mathbf{F}_2)$.*

2.4 The LWE Hardness Assumption

Let n be a positive integer dimension, let $q \geq 2$ be a prime, and let χ be a probability distribution over $\mathbb{Z}_q$. For $\boldsymbol{s} \in \mathbb{Z}_q^n$, let $A_{\boldsymbol{s},\chi}$ and $U_\$$ be two distributions defined as follows:

- $A_{\boldsymbol{s},\chi}$: the probability distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ obtained by choosing a random vector $\boldsymbol{a} \in \mathbb{Z}_q^n$ uniformly, choosing an error term $e \in \mathbb{Z}_q$ according to χ, and outputting $(\boldsymbol{a}, \langle \boldsymbol{a}, \boldsymbol{s} \rangle + e)$.
- $U_\$$: the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

For uniformly random $\boldsymbol{s} \in \mathbb{Z}_q^n$, an $(\mathbb{Z}_q, n, \chi)$-LWE problem instance consists of access to a challenge oracle $\mathcal{O}$ that outputs samples $(\boldsymbol{a}, b)$ from $\mathbb{Z}_q^n \times \mathbb{Z}_q$ according to, either the probability distribution $A_{\boldsymbol{s},\chi}$, or the uniform distribution $U_\$$. The $(\mathbb{Z}_q, n, \chi)$-LWE problem allows repeated queries to the challenge oracle $\mathcal{O}$. We say that an algorithm $\mathcal{A}$ decides the $\mathrm{LWE}_{\mathbb{Z}_q,n,\chi}$ problem if

$$\mathrm{Adv}_{\mathcal{A}}^{(\mathbb{Z}_q,n,\chi)\text{-LWE}} = |\Pr[\mathcal{A}^{\mathcal{O}_s} = 1] - \Pr[\mathcal{A}^{\mathcal{O}_\$} = 1]|$$

is non-negligible for a random $\boldsymbol{s} \in \mathbb{Z}_q^n$, where $\mathcal{O}_s$ and $\mathcal{O}_\$$ represent that the oracle $\mathcal{O}$ outputs samples from $\mathbb{Z}_q^n \times \mathbb{Z}_q$ according to $A_{\boldsymbol{s},\chi}$ and $U_\$$ respectively.

Regev [49] and Perkert [43] showed that for certain noise distributions χ, denoted $\bar{\Psi}_\alpha$, the LWE problem is as hard as the worst-case SIVP and GapSVP under a quantum reduction. Brakerski et al. [15] provided the first classical hardness reduction of LWE with polynomial modulus.

Definition 2 *Consider a real parameter $\alpha \in (0,1)$ and a prime q. Let $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ denote the group of reals $[0,1)$ with addition modulo 1. Let Ψ_α be the distribution on $\mathbb{T}$ obtained by sampling a normal variable with mean 0 and standard deviation $\alpha/\sqrt{2\pi}$ and reducing the result modulo 1. Let $\bar{\Psi}_\alpha$ denote the discrete distribution over $\mathbb{Z}_q$ of the random variable $\lfloor qX \rceil$ where the random variable $X \in \mathbb{T}$ has distribution Ψ_α.*

The following lemma about the distribution $\bar{\Psi}_\alpha$ taken from [1] will be needed to show that decryption works correctly.

Lemma 1 *Let $\boldsymbol{e}$ be some vector in $\mathbb{Z}^m$ and let $\boldsymbol{y} \leftarrow_R \bar{\Psi}_\alpha$. Then the quantity $|\boldsymbol{e}^\top \boldsymbol{y}|$ treated as an integer in $[0, q-1]$ satisfies $|\boldsymbol{e}^\top \boldsymbol{y}| \leq \|e\| q\alpha\omega(\sqrt{\log m}) + \|e\|\sqrt{m}/2$ with all but negligible probability in m.*

2.5 Vector Decomposition

Let k be an integer dimension, let $l = \lfloor \log_2 q \rfloor + 1$ and $N = k \cdot l$. Let $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{Z}_q^k$. We show a way of decomposing vectors that preserves the inner product [31]. We often break vectors into their bit representations as defined below:
$\mathsf{BitDecomp}(\boldsymbol{a})$: For $\boldsymbol{a} \in \mathbb{Z}_q^k$, let $a_{i,j}$ be the j-th bit in a_i's binary representation, bits ordered least significant to most significant. Output the N-dimensional vector $(a_{1,0}, \ldots, a_{1,l-1}, \ldots, a_{k,0}, \ldots, a_{k,l-1})$. $\mathsf{BitDecomp}^{-1}(\boldsymbol{a}')$: It is the inverse

of $\mathsf{BitDecomp}$. For $\boldsymbol{a}' = (a_{1,0}, \dots, a_{1,l-1}, \dots, a_{k,0}, \dots, a_{k,l-1})$, output $(\sum 2^j \cdot a_{1,j}, \dots, \sum 2^j \cdot a_{k,j})$. Note that, it is well-defined even if $\boldsymbol{a}'$ is not a 0/1 vector.

$\mathsf{Flatten}(\boldsymbol{a}')$: For N-dimensional vector $\boldsymbol{a}'$, output $\mathsf{BitDecomp}(\mathsf{BitDecomp}^{-1}(\boldsymbol{a}'))$, a N-dimensional vector with 0/1 coefficients.

$\mathsf{Powerof2}(\boldsymbol{b})$: For $\boldsymbol{b} = (b_1, \dots, b_k) \in \mathbb{Z}_q^k$, output the N-dimensional vector $(b_1, 2b_1, \dots, 2^{l-1}b_1, \dots, b_k, 2b_k, \dots, 2^{l-1}b_k)$.

Claim 1 *Let $\boldsymbol{a}, \boldsymbol{b}$ be vectors of some dimension k over $\mathbb{Z}_q$, let $\boldsymbol{a}'$ be any N-dimensional vector. We have*

- $\langle \mathsf{BitDecomp}(\boldsymbol{a}), \mathsf{Powerof2}(\boldsymbol{b}) \rangle = \langle \boldsymbol{a}, \boldsymbol{b} \rangle$.
- $\langle \boldsymbol{a}', \mathsf{Powerof2}(\boldsymbol{b}) \rangle = \langle \mathsf{BitDecomp}^{-1}(\boldsymbol{a}'), \boldsymbol{b} \rangle = \langle \mathsf{Flatten}(\boldsymbol{a}'), \mathsf{Powerof2}(\boldsymbol{b}) \rangle$.

When $\boldsymbol{A}$ is a matrix, let $\mathsf{BitDecomp}(\boldsymbol{A})$, $\mathsf{BitDecomp}^{-1}(\boldsymbol{A})$ or $\mathsf{Flatten}(\boldsymbol{A})$ be the matrix formed by applying the operation to each row of $\boldsymbol{A}$ separately.

2.6 Indistinguishability Obfuscation

Roughly speaking, an indistinguishability obfuscation ($i\mathcal{O}$) scheme ensures that the obfuscations of any two functionally equivalent circuits are computationally indistinguishable. Indistinguishability obfuscation was originally proposed by Barak et al. [6, 5] as a potential weakening of virtual-black-box obfuscation. We recall the definition from [27]. A uniform probabilistic polynomial time (PPT) machine $i\mathcal{O}$ is called an *indistinguishability obfuscator* for a circuit class $\{\mathcal{C}_\lambda\}_{\lambda \in \mathbb{N}}$ if the following conditions are satisfied:

- Correctness. For all security parameters $\lambda \in \mathbb{N}$, for all $C \in \mathcal{C}_\lambda$, and for all input x, we have that $\Pr[C'(x) = C(x) : C' \leftarrow i\mathcal{O}(\lambda, C)] = 1$.
- Security. For any (not necessarily uniform) PPT distinguisher D, for all pairs of circuits $C_0, C_1 \in \mathcal{C}_\lambda$ such that $C_0(x) = C_1(x)$ on all inputs x the following distinguishing advantage is negligible:

$$\mathsf{Adv}^{D}_{i\mathcal{O},C_0,C_1}(\lambda) := |\Pr[D(i\mathcal{O}(\lambda, C_0)) = 1] - \Pr[D(i\mathcal{O}(\lambda, C_1)) = 1]|.$$

2.7 Puncturable PRFs

A pseudorandom function (PRF) is a function $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ such that the function $F(K, \cdot)$ is indistinguishable from random when $K \leftarrow \mathcal{K}$. Puncturable PRFs were defined by Sahai and Waters [52] as a simple type of constrained PRFs [10, 12, 38]. They defined a puncturable PRF as a PRF for which a key can be given out that allows evaluation of the PRF on all inputs, except for a designated polynomial-size set of inputs. Formally, a *puncturable* PRF $F(K, \cdot)$ is equipped with additional PPT algorithms $(\mathsf{Eval}_F, \mathsf{Puncture}_F)$ such that the following properties hold:

- Correctness. For every PPT algorithm which on input a security parameter λ outputs a set $S \subseteq \{0,1\}^n$, for all $x \in \{0,1\}^n \backslash S$, we have that

$$\Pr[\mathsf{Eval}_F(K\{S\}, x) = F(K, x) : K \leftarrow \mathcal{K}, K\{S\} \leftarrow \mathsf{Puncture}_F(K, S)] = 1.$$

- Security. For any PPT algorithm $\mathcal{A}$, the following distinguishing advantage is negligible:

$$|\Pr[\mathcal{A}(S, K\{S\}, F(K, S)) = 1 : S \leftarrow \mathcal{A}(\lambda), K\{S\} \leftarrow \mathsf{Puncture}_F(K, S)] - \Pr[\mathcal{A}(S, K\{S\}, U_{\bar{\ell}\cdot|S|}) = 1 : S \leftarrow \mathcal{A}(\lambda), K\{S\} \leftarrow \mathsf{Puncture}_F(K, S)]|,$$

  where $F(K, S)$ denotes the concatenation of $F(K, x_1), \cdots, F(K, x_k)$, $S = \{x_1, \cdots, x_k\}$ is the enumeration of the elements of S in lexicographic order, $\bar{\ell}$ denotes the bit-length of the output $F(K, x)$, and U_ℓ denotes the uniform distribution over ℓ bits.

2.8 Keyed-Fully Homomorphic Encryption

A keyed-fully homomorphic encryption scheme consists of the following four algorithms:

$\mathsf{Setup}(1^\kappa)$ takes as input a security parameter κ. It outputs a public key PK, a decryption key DK and an evaluation key EK.

$\mathsf{Enc}(\mathsf{PK}, b)$ takes as input a public key PK and a message bit $b \in \{0,1\}$. It outputs a ciphertext C.

$\mathsf{Dec}(\mathsf{PK}, \mathsf{DK}, C)$ takes as input a public key PK, a decryption key DK and a ciphertext C. It outputs a message bit b.

$\mathsf{Eval}(\mathsf{PK}, \mathsf{EK}, \boldsymbol{C}, f)$ takes as input a public key PK, an evaluation key EK, a tuple of ciphertexts $\boldsymbol{C} = (C_1, \ldots, C_k)$ and a Boolean circuit $f : \{0,1\}^k \to \{0,1\}$. It outputs a ciphertext C.

Correctness. We require that for each $(\mathsf{PK}, \mathsf{DK}, \mathsf{EK})$ output by $\mathsf{Setup}(1^\kappa)$, the following hold:

Encryption correctness: with overwhelming probability, for all message bit $b \in \{0,1\}$, we have $\mathsf{Dec}(\mathsf{PK}, \mathsf{DK}, \mathsf{Enc}(\mathsf{PK}, b)) = b$.

Evaluation correctness: for any k-ciphertexts $(C_1, \ldots, C_k)$ such that $\mathsf{Dec}(\mathsf{PK}, \mathsf{DK}, C_i) = b_i \in \{0,1\}$, and a Boolean circuit $f : \{0,1\}^k \to \{0,1\}$, with overwhelming probability, we have

$$\mathsf{Dec}(\mathsf{PK}, \mathsf{DK}, \mathsf{Eval}(\mathsf{PK}, \mathsf{EK}, \boldsymbol{C} = (C_1, \ldots, C_k), f)) = f(b_1, \ldots, b_k).$$

Security. The CCA security of keyed-FHE scheme is defined using the following game between a PPT 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 [25]. That is, in our model, a keyed-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.

Setup The challenger runs $\mathsf{Setup}(1^\lambda)$ to obtain a public key PK, a decryption key DK and an evaluation key EK. It sends the public key PK to the adversary $\mathcal{A}$. In addition, the challenger maintains a list DList, which is set as $\emptyset$ initially.

Query phase 1 The adversary $\mathcal{A}$ adaptively issues the following queries:

- **DecCT**$\langle C \rangle$: The challenger uses the decryption key DK to decrypt C with algorithm Dec. The result is sent back to $\mathcal{A}$. *This query is not allowed to issue if $\mathcal{A}$ has queried to **RevEK**.*
- **EvalOnCT**$\langle \boldsymbol{C} = (C_1, \ldots, C_k), f \rangle$: The challenger runs $\mathsf{Eval}(\mathsf{PK}, \mathsf{EK}, \boldsymbol{C}, f)$ to obtain a ciphertext C, which is returned to $\mathcal{A}$. *This query is not allowed to issue if $\mathcal{A}$ has queried to **RevEK**.*
- **RevEK**: The challenger sends the evaluation key EK to $\mathcal{A}$.

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

Query phase 2 The adversary $\mathcal{A}$ continues to adaptively issue the following queries:

- **DecCT**$\langle C \rangle$: If $C \in \mathsf{DList}$, the challenger returns $\perp$. Otherwise, the challenger uses the decryption key DK to decrypt C with algorithm Dec, and the result is sent back to $\mathcal{A}$. *This query is not allowed to issue if $\mathcal{A}$ has queried to **RevEK**.*
- **EvalOnCT**$\langle \boldsymbol{C} = (C_1, \ldots, C_k), f \rangle$: The challenger runs $\mathsf{Eval}(\mathsf{PK}, \mathsf{EK}, \boldsymbol{C}, f)$ to obtain a ciphertext C, which is returned to $\mathcal{A}$. In addition, if there exists $i \in [k]$ such that $C_i \in \mathsf{DList}$, then the challenger updates the list by $\mathsf{DList} \leftarrow \mathsf{DList} \cup \{C\}$. *This query is not allowed to issue if $\mathcal{A}$ has queried to **RevEK**.*
- **RevEK**: The challenger sends the evaluation key EK to $\mathcal{A}$.

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 $|\mathsf{Pr}[b = b^*] - \frac{1}{2}|$ where the probability is taken over the random bits used by the challenger and the adversary.

Definition 3 *A keyed-FHE scheme is CCA-secure if all probabilistic polynomial time adversaries have at most a negligible advantage in the above security game.*

3 Convertible Identity-based Fully Homomorphic Encryption

Informally, a convertible IBFHE is an IBFHE with an additional transformation functionality: given a transformation key $\mathsf{TK}_{\mapsto \widetilde{\mathsf{ID}}}$ for an identity $\widetilde{\mathsf{ID}}$, which is generated by the authority, one can transform a ciphertext CT under *any* identity into a ciphertext under identity $\widetilde{\mathsf{ID}}$ without changing the underlying plaintext of CT. Concretely, a convertible IBFHE scheme consists of the following seven algorithms:

$\mathsf{Setup}(1^\kappa)$ takes as input a security parameter κ. It generates a public parameters PP and a master key MK.

$\mathsf{Extract}(\mathsf{PP}, \mathsf{MK}, \mathsf{ID})$ takes as input the public parameters PP, the master key MK and an identity ID. It produces a private key $\mathsf{SK}_{\mathsf{ID}}$ for identity ID.

$\mathsf{GenerateTK}(\mathsf{PP}, \mathsf{MK}, \widetilde{\mathsf{ID}})$ takes as input the public parameters PP, the master key MK and an identity $\widetilde{\mathsf{ID}}$. It generates a transformation key $\mathsf{TK}_{\mapsto \widetilde{\mathsf{ID}}}$ for identity $\widetilde{\mathsf{ID}}$.

$\mathsf{Encrypt}(\mathsf{PP}, \mathsf{ID}, b)$ takes as input the public parameters PP, an identity ID and a message bit $b \in \{0, 1\}$. It outputs a ciphertext CT.

$\mathsf{Transform}(\mathsf{PP}, \mathsf{TK}_{\mapsto \widetilde{\mathsf{ID}}}, \mathsf{ID}, \mathsf{CT})$ takes as input the public parameters PP, a transformation key $\mathsf{TK}_{\mapsto \widetilde{\mathsf{ID}}}$, and a ciphertext CT for an identity ID. It outputs a ciphertext $\widetilde{\mathsf{CT}}$ under identity $\widetilde{\mathsf{ID}}$.

$\mathsf{Decrypt}(\mathsf{PP}, \mathsf{SK}_{\mathsf{ID}}, \mathsf{CT})$ takes as input the public parameters PP, a private key $\mathsf{SK}_{\mathsf{ID}}$ and a ciphertext CT. It outputs a message bit $b \in \{0, 1\}$.

$\mathsf{Evaluate}(\mathsf{PP}, \mathsf{ID}, \boldsymbol{CT}, f)$ takes as input the public parameters PP, a tuple of ciphertexts $\boldsymbol{CT} = (\mathsf{CT}_1, \ldots, \mathsf{CT}_k)$ under an identity ID and a Boolean circuit $f : \{0, 1\}^k \to \{0, 1\}$. It outputs a ciphertext CT under identity ID.

Correctness. We require that for each $(\mathsf{PP}, \mathsf{MK})$ output by $\mathsf{Setup}(1^\kappa)$, the following hold:

Encryption correctness: with overwhelming probability, for all identity ID and message bit $b \in \{0, 1\}$, we have $\mathsf{Decrypt}(\mathsf{PP}, \mathsf{Extract}(\mathsf{PP}, \mathsf{MK}, \mathsf{ID}), \mathsf{Encrypt}(\mathsf{PP}, \mathsf{ID}, b)) = b$.

Transformation correctness: with overwhelming probability, for all identity $\mathsf{ID}, \widetilde{\mathsf{ID}}$ and message bit $b \in \{0, 1\}$, let $\mathsf{CT} \leftarrow \mathsf{Encrypt}(\mathsf{PP}, \mathsf{ID}, b)$, $\mathsf{SK}_{\mathsf{ID}} \leftarrow \mathsf{Extract}(\mathsf{PP}, \mathsf{MK}, \mathsf{ID})$, $\mathsf{SK}_{\widetilde{\mathsf{ID}}} \leftarrow \mathsf{Extract}(\mathsf{PP}, \mathsf{MK}, \widetilde{\mathsf{ID}})$, $\mathsf{TK}_{\mapsto \widetilde{\mathsf{ID}}} \leftarrow \mathsf{GenerateTK}(\mathsf{PP}, \mathsf{MK}, \widetilde{\mathsf{ID}})$, and $\widetilde{\mathsf{CT}} \leftarrow \mathsf{Transform}(\mathsf{PP}, \mathsf{TK}_{\mapsto \widetilde{\mathsf{ID}}}, \mathsf{ID}, \mathsf{CT})$, we have

$$\mathsf{Decrypt}(\mathsf{PP}, \mathsf{SK}_{\mathsf{ID}}, \mathsf{CT}) = \mathsf{Decrypt}(\mathsf{PP}, \mathsf{SK}_{\widetilde{\mathsf{ID}}}, \widetilde{\mathsf{CT}}).$$

Evaluation correctness: for any k-ciphertexts $(\mathsf{CT}_1, \ldots, \mathsf{CT}_k)$ under an identity ID such that $\mathsf{Decrypt}(\mathsf{PP}, \mathsf{Extract}(\mathsf{PP}, \mathsf{MK}, \mathsf{ID}), \mathsf{CT}_i) = b_i \in \{0, 1\}$, and a Boolean circuit $f : \{0, 1\}^k \to \{0, 1\}$, with overwhelming probability, we have $\mathsf{Decrypt}(\mathsf{PP}, \mathsf{Extract}(\mathsf{PP}, \mathsf{MK}, \mathsf{ID}), \mathsf{Evaluate}(\mathsf{PP}, \mathsf{ID}, \boldsymbol{CT} = (\mathsf{CT}_1, \ldots, \mathsf{CT}_k), f)) = f(b_1, \ldots, b_k)$.

Security. The $\mathsf{IND\text{-}sID\text{-}CPA}$ security of convertible IBFHE scheme is defined using the following game between a PPT adversary $\mathcal{A}$ and a challenger.

Init The adversary submits a target identity ID^* and a designated identity $\widetilde{\mathsf{ID}}$.

Setup The challenger first runs $\mathsf{Setup}(1^\kappa)$ to obtain a public parameters PP and a master key MK. Then, it runs $\mathsf{GenerateTK}(\mathsf{PP}, \mathsf{MK}, \widetilde{\mathsf{ID}})$ to get the transformation key $\mathsf{TK}_{\mapsto \widetilde{\mathsf{ID}}}$ for identity $\widetilde{\mathsf{ID}}$, and sends the public parameters PP and the transformation key $\mathsf{TK}_{\mapsto \widetilde{\mathsf{ID}}}$ to the adversary $\mathcal{A}$.

Query phase 1 The adversary $\mathcal{A}$ adaptively issues the following queries:

- **GetSK**$\langle\mathsf{ID}\rangle$: The challenger runs $\mathsf{Extract}(\mathsf{PP}, \mathsf{MK}, \mathsf{ID})$ to generate the corresponding private key $\mathsf{SK}_{\mathsf{ID}}$, which is returned to $\mathcal{A}$. We require that $\mathsf{ID} \notin \{\mathsf{ID}^*, \widetilde{\mathsf{ID}}\}$.

Challenge The challenger first selects a message bit $b^* \in \{0, 1\}$ uniformly at random. Then, it computes $\mathsf{CT}^* \leftarrow \mathsf{Encrypt}(\mathsf{PP}, \mathsf{ID}^*, b^*)$, and sends the challenge ciphertext CT^* to the adversary.

Query phase 2 This is same as Query phase 1.

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 $|\mathsf{Pr}[b = b^*] - \frac{1}{2}|$, where the probability is taken over the random bits used by the challenger and the adversary.

Definition 4 *A convertible IBFHE scheme is IND-sID-CPA secure, if the advantage in the above security game is negligible for all PPT adversaries.*

4 Proposed CCA Secure Keyed-FHE Scheme

Given a convertible IBFHE scheme $\mathsf{cIBE} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{GenerateTK}, \mathsf{Encrypt}, \mathsf{Transform}, \mathsf{Decrypt}, \mathsf{Evaluate})$ for identities of length ℓ which is IND-sID-CPA secure, we construct a CCA-secure keyed-FHE scheme. In the construction, we use a strongly EUF-CMA secure signature scheme $\mathcal{S} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ in which the verification key output by Gen has length ℓ. The construction of our CCA-secure keyed-FHE scheme is described as follows.

$\mathsf{Setup}(1^\kappa)$: The setup algorithm first runs $\mathsf{cIBE.Setup}(1^\kappa)$ to obtain $(\mathsf{PP}, \mathsf{MK})$, and calls $\mathcal{S}.\mathsf{Gen}(1^\kappa)$ to obtain a key pair $(\widetilde{vk}, \widetilde{sk})$. Then, it computes

$$\mathsf{TK}_{\mapsto \widetilde{vk}} \leftarrow \mathsf{cIBE.GenerateTK}(\mathsf{PP}, \mathsf{MK}, \widetilde{vk}).$$

Finally, it sets the public key $\mathsf{PK} = \mathsf{PP}$, the decryption key $\mathsf{DK} = \mathsf{MK}$ and the evaluation key $\mathsf{EK} = (\widetilde{vk}, \widetilde{sk}, \mathsf{TK}_{\mapsto \widetilde{vk}})$.

$\mathsf{Enc}(\mathsf{PK}, b \in \{0, 1\})$: The encryption algorithm takes as input the public key $\mathsf{PK} = \mathsf{PP}$, and a message bit $b \in \{0, 1\}$. It proceeds as follows.

1. Run $\mathcal{S}.\mathsf{Gen}(1^\kappa)$ to obtain a key pair (vk, sk).
2. Compute $\mathsf{CT} \leftarrow \mathsf{cIBE.Encrypt}(\mathsf{PP}, vk, b)$ and $\sigma \leftarrow \mathcal{S}.\mathsf{Sign}(sk, \mathsf{CT})$.
3. Output the ciphertext $C = (vk, \mathsf{CT}, \sigma)$.

$\mathsf{Dec}(\mathsf{PK}, \mathsf{DK}, C)$: The decryption algorithm takes as input the public key $\mathsf{PK} = \mathsf{PP}$, the decryption key $\mathsf{DK} = \mathsf{MK}$ and a ciphertext $C = (vk, \mathsf{CT}, \sigma)$. This algorithm first checks whether $\mathcal{S}.\mathsf{Vrfy}(vk, \mathsf{CT}, \sigma) = 1$. If not, it outputs $\perp$. Otherwise, it computes $\mathsf{SK}_{vk} \leftarrow \mathsf{cIBE.Extract}(\mathsf{PP}, \mathsf{MK}, vk)$ and sets $b \leftarrow \mathsf{cIBE.Decrypt}(\mathsf{PP}, \mathsf{SK}_{vk}, \mathsf{CT})$. Then, it outputs the message bit b.

$\mathsf{Eval}(\mathsf{PK}, \mathsf{EK}, \boldsymbol{C}, f)$: This algorithm takes as input the public key $\mathsf{PK} = \mathsf{PP}$, the evaluation key $\mathsf{EK} = (\widetilde{vk}, \widetilde{sk}, \mathsf{TK}_{\mapsto \widetilde{vk}})$, a tuple of ciphertexts $\boldsymbol{C} = (C_1 = (vk_1, \mathsf{CT}_1, \sigma_1), \ldots, C_k = (vk_k, \mathsf{CT}_k, \sigma_k))$ and a Boolean circuit $f : \{0,1\}^k \rightarrow \{0,1\}$. For $i = 1, \ldots, k$, it proceeds as follows.

1. Check whether $\mathcal{S}.\mathsf{Vrfy}(vk_i, \mathsf{CT}_i, \sigma_i) = 1$. If not, it outputs $\perp$.
2. Compute $\widetilde{\mathsf{CT}}_i \leftarrow \mathsf{cIBE.Transform}(\mathsf{PP}, \mathsf{TK}_{\mapsto \widetilde{vk}}, vk_i, \mathsf{CT}_i)$.

Next, it calls $\mathsf{cIBE.Evaluate}$ to obtain $\widetilde{\mathsf{CT}} \leftarrow \mathsf{cIBE.Evaluate}(\mathsf{PP}, \widetilde{vk}, (\widetilde{\mathsf{CT}}_1, \ldots, \widetilde{\mathsf{CT}}_k), f)$. Then, it computes $\tilde{\sigma} \leftarrow \mathcal{S}.\mathsf{Sign}(\widetilde{sk}, \mathsf{CT})$, and outputs the ciphertext $C = (\widetilde{vk}, \widetilde{\mathsf{CT}}, \tilde{\sigma})$.

Correctness. If the underlying convertible IBFHE scheme cIBE satisfies encryption correctness, transformation correctness and evaluation correctness, it is obvious that the above construction satisfies the correctness requirements of keyed-FHE.

Theorem 4 *If the underlying convertible IBFHE scheme is* IND-sID-CPA *secure, and the signature scheme* $\mathcal{S}$ *is strongly* EUF-CMA *secure, then our proposed keyed-FHE scheme is* CCA*-secure.*

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

Game 0. This is the original CCA security game between an adversary $\mathcal{A}$ against our scheme and a CCA challenger.

Game 1. In this game, we slightly change the way that the challenger answers the adversary's **DecCT** and **EvalOnCT** queries. Let $C^* = (vk^*, \mathsf{CT}^*, \sigma^*)$ be the challenge ciphertext.

When the adversary $\mathcal{A}$ issues a **DecCT** query on ciphertext $C = (vk, \mathsf{CT}, \sigma)$, the challenger checks whether $vk = vk^*, C \neq C^*$ and $\mathcal{S}.\mathsf{Vrfy}(vk, \mathsf{CT}, \sigma) = 1$. If so, the challenger returns $\perp$; otherwise, it responds as in Game 0.

When the adversary $\mathcal{A}$ issues an **EvalOnCT** query on $\langle \boldsymbol{C} = (C_1, \ldots, C_k), f \rangle$, the challenger first parses C_i as $(vk_i, \mathsf{CT}_i, \sigma_i)$ for each $i \in [k]$. Then, the challenger checks whether there exists $i \in [k]$ such that $vk_i = vk^*, C_i \neq C^*$ and $\mathcal{S}.\mathsf{Vrfy}(vk_i, \mathsf{CT}_i, \sigma_i) = 1$. If so, the challenger returns $\perp$; otherwise, it responds as in Game 0.

Game 2. In this game, at the setup phase, except for the list DList, the challenger also maintains another list EList, which is set as $\emptyset$ initially. We also modify the way how the adversary $\mathcal{A}$'s **DecCT** and **EvalOnCT** queries are answered. Let $\mathsf{PK}, \mathsf{DK}, \mathsf{EK} = (\widetilde{vk}, \widetilde{sk}, \mathsf{TK}_{\mapsto \widetilde{vk}})$ be the public key, decryption key and evaluation key respectively, generated by the challenger at the setup phase.

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

1. Check whether $\mathcal{S}.\mathsf{Vrfy}(vk, \mathsf{CT}, \sigma) = 1$. If not, return $\perp$.
2. Search the list EList for a record (m, C). If such record does not exist, return $\perp$; otherwise, send m to $\mathcal{A}$.

When the adversary $\mathcal{A}$ issues an **EvalOnCT** query on $\langle \boldsymbol{C} = (C_1, \ldots, C_k), f\rangle$, the challenger first parses C_i as $(vk_i, \mathsf{CT}_i, \sigma_i)$ for each $i \in [k]$. Then, it checks whether there exists $i \in [k]$ such that one of the following conditions holds: 1) $vk_i = vk^*, \mathcal{S}.\mathsf{Vrfy}(vk_i, \mathsf{CT}_i, \sigma_i) = 1$ and $C_i \neq C^*$; 2) $vk_i = \widetilde{vk}, \mathcal{S}.\mathsf{Vrfy}(vk_i, \mathsf{CT}_i, \sigma_i) = 1$ and the list EList does not contain a record (m_i, C_i). If so, the challenger returns $\perp$ to $\mathcal{A}$; otherwise, the challenger runs $\mathsf{Eval}(\mathsf{PK}, \mathsf{EK}, \boldsymbol{C}, f)$ 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 $C_i \in \mathsf{DList}$. If so, the challenger updates the list by $\mathsf{DList} \leftarrow \mathsf{DList} \cup \{C\}$; otherwise, it proceeds as follows.

1. For each $i \in [k]$, if $vk_i = \widetilde{vk}$, the challenger finds the record (m_i, C_i) in the list EList; otherwise (i.e., $vk_i \neq \widetilde{vk}$), the challenger uses the decryption key DK to decrypt C_i with algorithm Dec and obtain a message bit m_i.
2. The challenger computes $m = f(m_1, \ldots, m_k)$ and updates the list by $\mathsf{EList} \leftarrow \mathsf{EList} \cup \{(m, C)\}$.

By the following lemmas, we prove these games are computationally indistinguishable, and in Game 2, the advantage of the adversary is negligible. 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 4.

Lemma 2 *Suppose that the signature scheme $\mathcal{S}$ is strongly EUF-CMA-secure. Then Game 0 and Game 1 are computationally indistinguishable.*

Proof. Let $C^* = (vk^*, \mathsf{CT}^*, \sigma^*)$ be the challenge ciphertext. Define event E: the adversary $\mathcal{A}$ submits a ciphertext $C = (vk, \mathsf{CT}, \sigma)$ such that $vk = vk^*, C \neq C^*$ and $\mathcal{S}.\mathsf{Vrfy}(vk, \mathsf{CT}, \sigma) = 1$ during its **DecCT** or **EvalOnCT** queries. If E does not happen, Game 0 is identical to Game 1. All we have to do is to prove that E happens with negligible probability.

Suppose that E happens with non-negligible probability. Then we can build an algorithm $\mathcal{B}$ that breaks strong $\mathsf{EUF\text{-}CMA}$ security of the signature scheme $\mathcal{S}$ with non-negligible probability. Let $\mathcal{C}$ be the challenger corresponding to $\mathcal{B}$ in the strong $\mathsf{EUF\text{-}CMA}$ security game of the signature scheme $\mathcal{S}$. $\mathcal{B}$ is given the verification key vk^* of the signature scheme $\mathcal{S}$, and simulates Game 1 to the adversary $\mathcal{A}$ as follows.

$\mathcal{B}$ runs Setup to obtain $(\mathsf{PK}, \mathsf{DK}, \mathsf{EK})$, and sends the public key PK to $\mathcal{A}$. Since $\mathcal{B}$ knows the decryption key DK and the evaluation key EK associated with PK, thus it is able to answer all queries made by the adversary. At some point, $\mathcal{A}$ asks for the challenge ciphertext. $\mathcal{B}$ proceeds as follows.

1. Choose a message bit $b^* \in \{0, 1\}$ uniformly at random.
2. Compute $\mathsf{CT}^* \leftarrow \mathsf{cIBE.Encrypt}(\mathsf{PK}, vk^*, b^*)$.
3. Issue the signing query on CT^* to its challenger $\mathcal{C}$ to obtain the corresponding signature σ^*.

4. Set the challenge ciphertext $C^* = (vk^*, \mathsf{CT}^*, \sigma^*)$ and send it to the adversary $\mathcal{A}$.

Suppose E happens during the simulation (i.e., the adversary submits a ciphertext $C = (vk, \mathsf{CT}, \sigma)$ such that $vk = vk^*, C \neq C^*$ and $\mathcal{S}.\mathsf{Vrfy}(vk, \mathsf{CT}, \sigma) = 1$ during its **DecCT** or **EvalOnCT** queries), $\mathcal{B}$ outputs (CT, σ) which is not equal to $(\mathsf{CT}^*, \sigma^*)$, as its forgery of the signature scheme $\mathcal{S}$. Thus, if E happens with non-negligible probability, then $\mathcal{B}$ can break strong EUF-CMA security of the signature scheme $\mathcal{S}$ with non-negligible probability.

Lemma 3 *Suppose that the signature scheme $\mathcal{S}$ is strongly EUF-CMA-secure. Then Game 1 and Game 2 are computationally indistinguishable.*

Proof. Let $\mathsf{EK} = (\widetilde{vk}, \widetilde{sk}, \mathsf{TK}_{\mapsto \widetilde{vk}})$ be the evaluation key. Game 2 is the same as Game 1 except for the way of answering the adversary $\mathcal{A}$'s **DecCT** and **EvalOnCT** queries when $\mathcal{A}$ submits a ciphertext $C = (vk, \mathsf{CT}, \sigma)$ such that $vk = \widetilde{vk}$ and $\mathcal{S}.\mathsf{Vrfy}(vk, \mathsf{CT}, \sigma) = 1$. Recall that in our security definition of keyed-FHE, the adversary cannot issue the decryption or evaluation queries if it requests the evaluation key to be exposed. Since our proposed scheme satisfies the requirement of evaluation correctness, it is easy to observe that when $\mathcal{A}$ submits a ciphertext $C = (vk = \widetilde{vk}, \mathsf{CT}, \sigma)$ during its **DecCT** or **EvalOnCT** queries such that C is the return of $\mathcal{A}$'s some **EvalOnCT** query, the challenger's response is the same in Game 1 and Game 2.

Define event E: the adversary $\mathcal{A}$ submits a ciphertext $C = (vk = \widetilde{vk}, \mathsf{CT}, \sigma)$ during its **DecCT** or **EvalOnCT** queries such that $\mathcal{S}.\mathsf{Vrfy}(vk, \mathsf{CT}, \sigma) = 1$ and C is not the response to $\mathcal{A}$'s some **EvalOnCT** query. If E does not happen, Game 1 is identical to Game 2. All we have to do is to prove that E happens with negligible probability.

One can prove that if the signature scheme $\mathcal{S}$ is strongly EUF-CMA-secure, then event E happens with negligible probability. We omit the details due to its similarity of Lemma 2.

Lemma 4 *If the underlying convertible IBFHE scheme is IND-sID-CPA-secure, then in Game 2, the advantage of the adversary is negligible.*

Proof. Suppose there exists 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 cIBE in the IND-sID-CPA security game with a non-negligible advantage. Let $\mathcal{C}$ be the challenger corresponding to $\mathcal{B}$ in the IND-sID-CPA security game of the convertible IBFHE scheme cIBE. $\mathcal{B}$ runs $\mathcal{A}$ executing the following steps.

Setup $\mathcal{B}$ first runs $\mathcal{S}.\mathsf{Gen}$ twice to obtain two key pairs (vk^*, sk^*) and $(\widetilde{vk}, \widetilde{sk})$. Then, it submits $(vk^*, \widetilde{vk})$ to $\mathcal{C}$ as its target identity and designated identity, and $\mathcal{C}$ returns the public parameters PP of the convertible IBFHE scheme cIBE and the transformation key $\mathsf{TK}_{\mapsto \widetilde{vk}}$ for identity $\widetilde{vk}$ to $\mathcal{B}$. Next, $\mathcal{B}$ sets the public key $\mathsf{PK} = \mathsf{PP}$, the evaluation key $\mathsf{EK} = (\widetilde{vk}, \widetilde{sk}, \mathsf{TK}_{\mapsto \widetilde{vk}})$, and sends the public key PK to the adversary $\mathcal{A}$. In addition, $\mathcal{B}$ maintains two lists DList and EList, which are set as $\emptyset$ initially.

Query phase 1 The adversary $\mathcal{A}$ adaptively issues the following queries:

- **DecCT**$\langle C \rangle$: $\mathcal{B}$ first parses the ciphertext C as $(vk, \mathsf{CT}, \sigma)$. Then, it checks whether $\mathcal{S}.\mathsf{Vrfy}(vk,\ \mathsf{CT}, \sigma) = 1$. If not, $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$; otherwise, $\mathcal{B}$ proceeds as follows.
  1. If $vk = vk^*$, $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$.
  2. If $vk \neq \widetilde{vk}$, $\mathcal{B}$ issues **GetSK** query on $\langle vk \rangle$ to its challenger $\mathcal{C}$ to obtain a private key SK_{vk} for identity vk, and uses the private key SK_{vk} to decrypt CT with algorithm $\mathsf{cIBE.Decrypt}$. The result is sent back to $\mathcal{A}$.
  3. If $vk = \widetilde{vk}$ and $C \in \mathsf{DList}$, $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$.
  4. Otherwise (i.e., $vk = \widetilde{vk}$ and $C \notin \mathsf{DList}$), $\mathcal{B}$ search the list EList for a record (m, C). If such record does not exist, $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$; otherwise, $\mathcal{B}$ sends the message bit m to the adversary $\mathcal{A}$.
- **EvalOnCT**$\langle \boldsymbol{C} = (C_1, \ldots, C_k), f \rangle$: For each $i \in [k]$, $\mathcal{B}$ parses C_i as $(vk_i, \mathsf{CT}_i, \sigma_i)$. Then, $\mathcal{B}$ checks whether there exists $i \in [k]$ such that one of the following conditions holds: 1) $vk_i = vk^*, \mathcal{S}.\mathsf{Vrfy}(vk_i, \mathsf{CT}_i, \sigma_i) = 1$ and $C_i \neq C^*$; 2) $vk_i = \widetilde{vk}, \mathcal{S}.\mathsf{Vrfy}(vk_i, \mathsf{CT}_i, \sigma_i) = 1$ and the list EList does not contain a record (m_i, C_i). If so, $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$; otherwise, $\mathcal{B}$ runs $\mathsf{Eval}(\mathsf{PK}, \mathsf{EK}, \boldsymbol{C}, f)$ to obtain a ciphertext C, which is returned to $\mathcal{A}$. In addition, when the ciphertext $C \neq \perp$, $\mathcal{B}$ checks whether there exists $i \in [k]$ such that $C_i \in \mathsf{DList}$. If so, $\mathcal{B}$ updates the list by $\mathsf{DList} \leftarrow \mathsf{DList} \cup \{C\}$; otherwise, $\mathcal{B}$ proceeds as follows.
  1. For each $i \in [k]$, if $vk_i = \widetilde{vk}$, $\mathcal{B}$ finds the record (m_i, C_i) in the list EList; otherwise (i.e., $vk_i \neq \widetilde{vk}$), $\mathcal{B}$ issues **GetSK** query on $\langle vk_i \rangle$ to its challenger $\mathcal{C}$ to obtain a private key SK_{vk_i} for identity vk_i, and uses the private key SK_{vk_i} to decrypt CT_i with algorithm $\mathsf{cIBE.Decrypt}$ and obtain a message bit m_i.
  2. $\mathcal{B}$ computes $m = f(m_1, \ldots, m_k)$ and updates the list by $\mathsf{EList} \leftarrow \mathsf{EList} \cup \{(m, C)\}$.
- **RevEK**: The challenger sends the evaluation key EK to $\mathcal{A}$.

Challenge Firstly, $\mathcal{B}$ asks $\mathcal{C}$ for its challenge ciphertext of the convertible IBFHE scheme cIBE, and receives the ciphertext CT^*. Then, $\mathcal{B}$ computes $\sigma^* \leftarrow \mathcal{S}.\mathsf{Sign}(sk^*, \mathsf{CT}^*)$, and sets the challenge ciphertext $C^* = (vk^*, \mathsf{CT}^*, \sigma^*)$. Finally, $\mathcal{B}$ sends C^* to the adversary $\mathcal{A}$. In addition, $\mathcal{B}$ updates the list by $\mathsf{DList} \leftarrow \mathsf{DList} \cup \{C^*\}$.

Query phase 2 The adversary $\mathcal{A}$ continues to adaptively issue **DecCT**, **EvalOnCT** and **RevEK** queries. $\mathcal{B}$ responds as Query phase 1.

Guess The adversary $\mathcal{A}$ outputs a bit $b \in \{0, 1\}$. $\mathcal{B}$ also takes b as its output.

It is easy to observe that, $\mathcal{B}$'s simulation is perfect. Thus, if $\mathcal{A}$ has a non-negligible advantage in Game 2, then $\mathcal{B}$ attacks the underlying convertible IBFHE scheme cIBE in the $\mathsf{IND\text{-}sID\text{-}CPA}$ security game with a non-negligible advantage.

5 Proposed Convertible IBFHE Scheme

We denote $\mathsf{SampleUS}(\mathbb{Z}_q^n; r_S)$ as a sample algorithm that chooses an element in $\mathbb{Z}_q^n$ uniformly at random with the randomness r_S, $\mathsf{SampleGX}(\mathbb{Z}_q, \bar{\Psi}_\alpha; r_X)$ as a sample algorithm that chooses an element in $\mathbb{Z}_q$ from the distribution $\bar{\Psi}_\alpha$ with the randomness r_X, $\mathsf{SampleGY}(\mathbb{Z}_q^m, \bar{\Psi}_\alpha^m; r_Y)$ as a sample algorithm that chooses an element in $\mathbb{Z}_q^m$ from the distribution $\bar{\Psi}_\alpha^m$ with the randomness r_Y, and $\mathsf{SampleURs}(\{-1,1\}^{m\times m}; r_R)$ as a sample algorithm that chooses ℓ-elements in domain $\{-1,1\}^{m\times m}$ uniformly at random with the randomness r_R. Let $\mathsf{F}_D : \mathcal{K}_D \times \{0,1\}^* \to \{-1,1\}^\ell$, $\mathsf{F}_S : \mathcal{K}_S \times \{0,1\}^* \times \{0,1\}^{2\kappa} \times [N] \to \mathcal{R}_{\mathsf{SampleUS}}$, $\mathsf{F}_X : \mathcal{K}_X \times \{0,1\}^* \times \{0,1\}^{2\kappa} \times [N] \to \mathcal{R}_{\mathsf{SampleGX}}$, $\mathsf{F}_Y : \mathcal{K}_Y \times \{0,1\}^* \times \{0,1\}^{2\kappa} \times [N] \to \mathcal{R}_{\mathsf{SampleGY}}$, $\mathsf{F}_R : \mathcal{K}_R \times \{0,1\}^* \times \{0,1\}^{2\kappa} \times [N] \to \mathcal{R}_{\mathsf{SampleURs}}$ be puncturable PRFs, and let $\mathsf{PRG} : \{0,1\}^\kappa \to \{0,1\}^{2\kappa}$ be a pseudorandom generator. Let $i\mathcal{O}$ be a program indistinguishability obfuscator. Our proposed convertible identity-based (leveled) FHE scheme consists of the following algorithms:

$\mathsf{Setup}(1^\kappa, 1^L)$: On input a security parameter κ and a number of levels L (maximum circuit depth to support), this algorithm first chooses the parameters $(q, n, m, \sigma, \alpha)$ as specified in Section 5.1 below. Let $N = (2m+1)\cdot(\lfloor \log q \rfloor + 1)$. It then invokes $\mathsf{TrapGen}(q, n)$ to generate a uniformly random matrix $A \in \mathbb{Z}_q^{n\times m}$ and a short basis $T_A \in \mathbb{Z}^{m\times m}$ for $\Lambda_q^\perp(A)$. It also chooses uniformly random matrices $B_0, B_1, \ldots, B_\ell \in \mathbb{Z}_q^{n\times m}$, and a uniformly random vector $u \in \mathbb{Z}_q^n$. Next, it chooses puncturable PRF keys $\mathrm{K}_D \leftarrow \mathcal{K}_D, \mathrm{K}_S \leftarrow \mathcal{K}_S, \mathrm{K}_X \leftarrow \mathcal{K}_X, \mathrm{K}_Y \leftarrow \mathcal{K}_Y, \mathrm{K}_R \leftarrow \mathcal{K}_R$, and creates an obfuscation of the program $\mathsf{ProduceCT}$ as $\mathbf{P}^{\mathsf{Enc}} \leftarrow i\mathcal{O}(\kappa, \mathsf{ProduceCT})$,

```
ProduceCT:
Input: Identity $\mathsf{ID} \in \{0,1\}^*$, a message $b \in \{0,1\}$, and randomness $r \in \{0,1\}^\kappa$.
Constants: PRF keys $\mathrm{K}_D, \mathrm{K}_S, \mathrm{K}_X, \mathrm{K}_Y$ and $\mathrm{K}_R$.
  1. Compute $\mathsf{id} = \mathsf{F}_D(\mathrm{K}_D, \mathsf{ID}) \in \{-1,1\}^\ell$.
  2. Compute $t = \mathsf{PRG}(r)$.
  3. For each $i \in [N]$, do the following:
     (a) compute $r_{S,i} = \mathsf{F}_S(\mathrm{K}_S, \mathsf{ID}, t, i), r_{X,i} = \mathsf{F}_X(\mathrm{K}_X, \mathsf{ID}, t, i), r_{Y,i} = \mathsf{F}_Y(\mathrm{K}_Y, \mathsf{ID}, t, i)$ and $r_{R,i} = \mathsf{F}_R(\mathrm{K}_R, \mathsf{ID}, t, i)$;
     (b) evaluate $(c_{i,0}, c_{i,1}) = \mathsf{ABBEnc0}(\mathsf{PP}_{\mathsf{ABB}}, \mathsf{id}, r_{S,i}, r_{X,i}, r_{Y,i}, r_{R,i})$.
  4. Set $c_{\mathsf{id}} = \begin{bmatrix} c_{1,0}|c_{1,1}^\top \\ \vdots \\ c_{N,0}|c_{N,1}^\top \end{bmatrix} \in \mathbb{Z}_q^{N\times(2m+1)}$ and compute $C_{\mathsf{ID}} = \mathsf{Flatten}(b \cdot I_N + \mathsf{BitDecomp}(c_{\mathsf{id}}))$.
  5. Output: $(t, C_{\mathsf{ID}})$.
```

Fig. 1. Program $\mathsf{ProduceCT}$

Finally, it outputs the public parameters

$$\mathsf{PP} = \Big(\mathsf{PP}_{\mathsf{ABB}} = (A, B_0, B_1, \ldots, B_\ell, u), \mathbf{P}^{\mathsf{Enc}}, \mathsf{PRG}, \mathsf{F}_D, \mathsf{F}_S, \mathsf{F}_X, \mathsf{F}_Y, \mathsf{F}_R\Big),$$

and master key $\mathsf{MK} = \Big(T_A, \mathrm{K}_D, \mathrm{K}_S, \mathrm{K}_X, \mathrm{K}_Y, \mathrm{K}_R\Big)$.

$\mathsf{Extract}(\mathsf{PP}, \mathsf{MK}, \mathsf{ID})$: On input public parameters PP, a master key MK, and an identity $\mathsf{ID} \in \{0,1\}^*$, this algorithm first sets $\mathsf{id} = \mathsf{F}_D(\mathrm{K}_D, \mathsf{ID}) = (d_1, \ldots, d_\ell) \in \{-1,1\}^\ell$ and evaluates $\boldsymbol{e}_{\mathsf{ID}} \leftarrow \mathsf{SampleLeft}(A, B_0 + \sum_{i=1}^{\ell} d_i B_i, T_A, u, \sigma)$ to obtain a short vector in $\Lambda_q^u(F_{\mathsf{id}})$, where $F_{\mathsf{id}} = A \mid B_0 + \sum_{i=1}^{\ell} d_i B_i$ and $\boldsymbol{e}_{\mathsf{ID}}$ is distributed as $D_{\Lambda_q^u(F_{\mathsf{id}}),\sigma}$. Then, it sets $\boldsymbol{s}_{\mathsf{ID}} = (1, -\boldsymbol{e}_{\mathsf{ID}})$ and $\boldsymbol{v}_{\mathsf{ID}} = \mathsf{Powersof2}(\boldsymbol{s}_{\mathsf{ID}})$, and outputs the private key $\mathsf{SK}_{\mathsf{ID}} = \boldsymbol{v}_{\mathsf{ID}}$ for identity ID.

$\mathsf{GenerateTK}(\mathsf{PP}, \mathsf{MK}, \widetilde{\mathsf{ID}})$: On input public parameter PP, a master key MK, and an identity $\widetilde{\mathsf{ID}} \in \{0,1\}^*$, this algorithm creates an obfuscation of the program $\mathsf{ConvertTAID}$ as $\mathbf{P}^{\mathsf{Trans}} \leftarrow i\mathcal{O}(\kappa, \mathsf{ConvertTAID})$, and outputs the transformation key $\mathsf{TK}_{\mapsto \widetilde{\mathsf{ID}}} = \Big(\widetilde{\mathsf{ID}}, \mathbf{P}^{\mathsf{Trans}}\Big)$.

$\mathsf{Encrypt}(\mathsf{PP}, \mathsf{ID}, b)$: On input public parameters PP, an identity $\mathsf{ID} \in \{0,1\}^*$, and a message $b \in \{0,1\}$, the encryption algorithm first chooses $r \in \{0,1\}^\kappa$ uniformly at random. Then, it computes $(t, C_{\mathsf{ID}}) = \mathbf{P}^{\mathsf{Enc}}(\mathsf{ID}, b, r)$, and outputs the ciphertext $\mathsf{CT} = (t, C_{\mathsf{ID}})$.

$\mathsf{Transform}(\mathsf{PP}, \mathsf{TK}_{\mapsto \widetilde{\mathsf{ID}}}, \mathsf{ID}, \mathsf{CT})$: On input public parameters PP, a transformation key $\mathsf{TK}_{\mapsto \widetilde{\mathsf{ID}}} = (\widetilde{\mathsf{ID}}, \mathbf{P}^{\mathsf{Trans}})$, a ciphertext $\mathsf{CT} = (t, C_{\mathsf{ID}})$ for an identity ID, this algorithm computes $(\tilde{t}, C_{\widetilde{\mathsf{ID}}}) = \mathbf{P}^{\mathsf{Trans}}(\mathsf{ID}, \mathsf{CT})$, and outputs the transformed ciphertext $\widetilde{\mathsf{CT}} = (\tilde{t}, C_{\widetilde{\mathsf{ID}}})$.

$\mathsf{Decrypt}(\mathsf{PP}, \mathsf{SK}_{\mathsf{ID}}, \mathsf{CT})$: The decryption algorithm takes as input public parameters PP, a private key $\mathsf{SK}_{\mathsf{ID}} = \boldsymbol{v}_{\mathsf{ID}}$, and a ciphertext $\mathsf{CT} = (t, C_{\mathsf{ID}})$. Observe that the first $\lfloor \log q \rfloor + 1$ coefficients of $\boldsymbol{v}_{\mathsf{ID}}$ are $1, 2, \ldots, 2^{\lfloor \log q \rfloor}$. Among these coefficients, let $v_{\mathsf{ID},i} = 2^i$ be in $(q/4, q/2]$. Let $C_{\mathsf{ID},i}$ be the i-th row of C_{ID}. This algorithm computes $x_i \leftarrow \langle C_{\mathsf{ID},i}, \boldsymbol{v}_{\mathsf{ID}} \rangle$ and outputs $\mu' = \lfloor x_i / v_{\mathsf{ID},i} \rceil$.

$\mathsf{Evaluate}(\mathsf{PP}, \mathsf{ID}, \boldsymbol{CT}, f)$: The evaluation algorithm takes as input public parameters PP, a tuple of ciphertext $\boldsymbol{CT} = (\mathsf{CT}_1, \ldots, \mathsf{CT}_k)$ under an identity ID and a Boolean circuit $f : \{0,1\}^k \rightarrow \{0,1\}$. It is a remarkable fact that, Boolean circuits computed over encryptions of binary values can be converted to use only NAND gates [31]. Let $\mathsf{CT}_i = (t_i, ct_i)$ be an encryption of $b_i \in \{0,1\}$ for all $i \in [k]$, the NAND homomorphic operation is described below:
$\mathsf{NAND}(ct_1, ct_2)$: To NAND ciphertexts $ct_1, ct_2 \in \mathbb{Z}_q^{N \times N}$, output $\mathsf{Flatten}(I_N - ct_1 \cdot ct_2)$.
Let ct be the result of $f(ct_1, \ldots, ct_k)$ through appropriate leveled application of the NAND homomorphic operation. The algorithm chooses a random value $t \in \{0,1\}^{2\kappa}$ and outputs the resulting ciphertext $\mathsf{CT} = (t, ct)$.

$\mathsf{ABBEnc0}(\mathsf{PP}_{\mathsf{ABB}}, \mathsf{id}, r_S, r_X, r_Y, r_R)$: On input public parameters $\mathsf{PP}_{\mathsf{ABB}}$, an identity $\mathsf{id} = (d_1, \ldots, d_\ell) \in \{-1,1\}^\ell$ and randomness $r_S \in \mathcal{R}_{\mathsf{SampleUS}}, r_X \in$

```
ConvertTAID:
Input: Identity $\mathsf{ID} \in \{0,1\}^*$, and ciphertext $\mathsf{CT} = (t, C_{\mathsf{ID}}) \in \{0,1\}^{2\kappa} \times \mathbb{Z}_q^{N \times N}$.
Constants: PRF keys $\mathrm{K}_D, \mathrm{K}_S, \mathrm{K}_X, \mathrm{K}_Y$ and $\mathrm{K}_R$.
  1. If $\mathsf{ID}$ is equal to $\widetilde{\mathsf{ID}}$, output $\mathsf{CT}$.
  2. Compute $\mathsf{id} = \mathsf{F}_D(\mathrm{K}_D, \mathsf{ID}) \in \{-1,1\}^\ell$ and $\widetilde{\mathsf{id}} = \mathsf{F}_D(\mathrm{K}_D, \widetilde{\mathsf{ID}}) \in \{-1,1\}^\ell$.
  3. For each $i \in [N]$, do the following:
     (a) compute $r_{S,i} = \mathsf{F}_S(\mathrm{K}_S, \mathsf{ID}, t, i), r_{X,i} = \mathsf{F}_X(\mathrm{K}_X, \mathsf{ID}, t, i), r_{Y,i} = \mathsf{F}_Y(\mathrm{K}_Y, \mathsf{ID}, t, i)$ and $r_{R,i} = \mathsf{F}_R(\mathrm{K}_R, \mathsf{ID}, t, i)$;
     (b) evaluate $(c_{i,0}, c_{i,1}) = \mathsf{ABBEnc0}(\mathsf{PP}_{\mathsf{ABB}}, \mathsf{id}, r_{S,i}, r_{X,i}, r_{Y,i}, r_{R,i})$.
  4. Set $c_{\mathsf{id}} = \begin{bmatrix} c_{1,0} | c_{1,1}^\top \\ \vdots \\ c_{N,0} | c_{N,1}^\top \end{bmatrix} \in \mathbb{Z}_q^{N \times (2m+1)}$.
  5. Check whether there exists $b \in \{0,1\}$ such that $C_{\mathsf{ID}} = \mathsf{Flatten}(b \cdot I_N + \mathsf{BitDecomp}(c_{\mathsf{id}}))$.
     If not, output $\perp$.
  6. For each $i \in [N]$, do the following:
     (a) compute $\tilde{r}_{S,i} = \mathsf{F}_S(\mathrm{K}_S, \widetilde{\mathsf{ID}}, t, i), \tilde{r}_{X,i} = \mathsf{F}_X(\mathrm{K}_X, \widetilde{\mathsf{ID}}, t, i), \tilde{r}_{Y,i} = \mathsf{F}_Y(\mathrm{K}_Y, \widetilde{\mathsf{ID}}, t, i)$ and $\tilde{r}_{R,i} = \mathsf{F}_R(\mathrm{K}_R, \widetilde{\mathsf{ID}}, t, i)$;
     (b) evaluate $(\tilde{c}_{i,0}, \tilde{c}_{i,1}) = \mathsf{ABBEnc0}(\mathsf{PP}_{\mathsf{ABB}}, \widetilde{\mathsf{id}}, \tilde{r}_{S,i}, \tilde{r}_{X,i}, \tilde{r}_{Y,i}, \tilde{r}_{R,i})$.
  7. Set $c_{\widetilde{\mathsf{id}}} = \begin{bmatrix} \tilde{c}_{1,0} | \tilde{c}_{1,1}^\top \\ \vdots \\ \tilde{c}_{N,0} | \tilde{c}_{N,1}^\top \end{bmatrix} \in \mathbb{Z}_q^{N \times (2m+1)}$ and compute $C_{\widetilde{\mathsf{ID}}} = \mathsf{Flatten}(b \cdot I_N + \mathsf{BitDecomp}(c_{\widetilde{\mathsf{id}}}))$.
  8. Output: $(t, C_{\widetilde{\mathsf{ID}}})$.
```

Fig. 2. Program ConvertTAID

$\mathcal{R}_{\mathsf{SampleGX}}, r_Y \in \mathcal{R}_{\mathsf{SampleGY}}, r_R \in \mathcal{R}_{\mathsf{SampleURs}}$, this algorithm proceeds as follows.

1. Let $F_{\mathsf{id}} = A \mid B_0 + \sum_{i=1}^{\ell} d_i B_i \in \mathbb{Z}_q^{n \times 2m}$.
2. Evaluate $s = \mathsf{SampleUS}(\mathbb{Z}_q^n; r_S), x = \mathsf{SampleGX}(\mathbb{Z}_q, \bar{\Psi}_\alpha; r_X)$ and $y = \mathsf{SampleGY}(\mathbb{Z}_q^m, \bar{\Psi}_\alpha^m; r_Y)$.
3. Evaluate $(R_1, \ldots, R_\ell) = \mathsf{SampleURs}(\{-1,1\}^{m \times m}; r_R)$.
4. Set $R_{\mathsf{id}} = \sum_{i=1}^{\ell} d_i R_i$ and compute $z = R_{\mathsf{id}}^\top y$.
5. Compute $c_0 = u^\top s + x \in \mathbb{Z}_q$, $c_1 = F_{\mathsf{id}}^\top s + \begin{bmatrix} y \\ z \end{bmatrix} \in \mathbb{Z}_q^{2m}$, and return (c_0, c_1).

5.1 Parameters and Correctness

Let $\mathsf{CT}_1 = (t_1, ct_1 = \mathsf{Flatten}(b_1 \cdot I_N + \mathsf{BitDecomp}(c_1)))$ be an encryption of $b_1 \in \{0,1\}$ under an identity ID. Recall that c_1 is a N-row matrix whose rows are encryptions of 0 generated by using $\mathsf{ABBEnc0}$, and the private key $\boldsymbol{v}_{\mathsf{ID}} =$

$\mathsf{Powersof2}(\boldsymbol{s}_{\mathsf{ID}})$. By Claim 1, we have

$$ct_1 \cdot \boldsymbol{v}_{\mathsf{ID}} = (b_1 \cdot I_N + \mathsf{BitDecomp}(c_1)) \cdot \boldsymbol{v}_{\mathsf{ID}} = b_1 \cdot \boldsymbol{v}_{\mathsf{ID}} + c_1 \cdot \boldsymbol{s}_{\mathsf{ID}}.$$

Let $c_{1,i}$ be the i-th row of the matrix c_1, and let $v_{\mathsf{ID},i}$ be the i-th coefficient of $\boldsymbol{v}_{\mathsf{ID}}$. Algorithm Decrypt only uses the i-th coefficient of $ct_1 \cdot \boldsymbol{v}_{\mathsf{ID}}$, which is $x_i = b_1 \cdot v_{\mathsf{ID},i} + c_{1,i} \cdot \boldsymbol{s}_{\mathsf{ID}}$. If $c_{1,i}$ is properly generated using ABBEnc0, then the norm of the inner product $c_{1,i} \cdot \boldsymbol{s}_{\mathsf{ID}}$ is bounded *w.h.p* by $B = q\sigma\ell m\alpha\omega(\sqrt{\log m}) + O(\sigma m^{3/2})$ by Lemma 24 of [1]. Similarly as in [31], if $B < q/8$ and $v_{\mathsf{ID},i} \in (q/4, q/2]$, then $x_i/v_{\mathsf{ID},i}$ differs from b_1 by at most $(q/8)/v_{\mathsf{ID},i} < 1/2$, and algorithm Decrypt uses rounding to output the correct value of b_1.

It is clear that our system satisfies transformation correctness if encryption correctness holds. Regarding evaluation correctness, let $\mathsf{CT}_2 = (t_2, ct_2 = \mathsf{Flatten}(b_2 \cdot I_N + \mathsf{BitDecomp}(c_2)))$ be an encryption of another bit $b_2 \in \{0, 1\}$ under the same identity ID, where c_2 is also a N-row matrix whose rows are encryptions of 0 generated by using ABBEnc0. We have

$$\mathsf{NAND}(ct_1, ct_2) \cdot \boldsymbol{v}_{\mathsf{ID}} = (I_N - ct_1 \cdot ct_2) \cdot \boldsymbol{v}_{\mathsf{ID}} = (1 - b_1 b_2) \cdot \boldsymbol{v}_{\mathsf{ID}} - b_2 \cdot (c_1 \cdot \boldsymbol{s}_{\mathsf{ID}}) - ct_1 \cdot (c_2 \cdot \boldsymbol{s}_{\mathsf{ID}}).$$

Note that NAND maintains the invariant that if ct_1 and ct_2 are encryptions of messages in $\{0, 1\}$, then the output ciphertext is also encryption of message in $\{0, 1\}$. After an NAND homomorphic operation, the error is increased by a factor of at most $N + 1$.

Recall that we represent the homomorphic function f over encryptions of binary values as a Boolean circuit that can be converted to use only NAND gates. Through appropriate leveled application of the NAND homomorphic operation, the final ciphertext's error will be bounded by $(N+1)^L \cdot B$, where L is the NAND-depth of the circuit. As long as $(N + 1)^L \cdot B < q/8$, the decryption algorithm will decrypt correctly.

Hence, for the system to work correctly and evaluate a circuit of depth L, we set the parameters $(q, n, m, \sigma, \alpha)$ that satisfy the following requirements, taking n to be the security parameter κ:

- the error term has magnitude at most $q/8$ *w.h.p* (i.e. $B \cdot (N + 1)^L < q/8$),
- algorithm TrapGen can operate (i.e. $m > 6n \log q$),
- σ is sufficiently large for SampleLeft and SampleRight (i.e. $\sigma > \ell m\omega(\sqrt{\log m})$),
- Regev's reduction applies [49] (i.e $\alpha q > 2\sqrt{n}$),
- our security reduction applies (i.e. $q > 2Q + 4$, where Q is the number of private key queries from the adversary).

5.2 Security

We now state the security theorem of our proposed scheme.

Theorem 5 *If the $(\mathbb{Z}_q, n, \bar{\Psi}_\alpha)$-LWE assumptions holds, the proposed convertible IBFHE scheme is IND-sID-CPA secure.*

Proof. See the full version of this paper.

6 Conclusion

We introduced a new primitive called convertible IBFHE which is an IBFHE with an additional transformation functionality. We showed that CCA-secure keyed-FHE can be constructed from IND-sID-CPA-secure convertible IBFHE and strongly EUF-CMA-secure signature. Utilizing the recent advances in indistinguishability obfuscation, we presented a concrete construction of IND-sID-CPA-secure convertible IBFHE without random oracles, and yielded the first CCA-secure keyed-FHE scheme in the standard model. Since indistinguishability obfuscation is a slightly cumbersome primitive currently, thus it would be interesting to construct an efficient IND-sID-CPA-secure convertible IBFHE without using indistinguishability obfuscation, even in the random oracle model.

Acknowledgement

We are grateful to the anonymous reviewers for their helpful comments. The work of Junzuo Lai was supported by National Natural Science Foundation of China (Nos. 61572235, 61300226), Research Fund for the Doctoral Program of Higher Education of China (No. 20134401120017), Guangdong Natural Science Funds for Distinguished Young Scholar (No. 2015A030306045), Natural Science Foundation of Guangdong Province (No. 2014A030310156), Pearl River S&T Nova Program of Guangzhou and Fundamental Research Funds for the Central Universities (No. 21615445). The work of Jian Weng was supported by National Natural Science Foundation of China (Nos. 61272413, 61472165, 61133014).

References

1. S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*, pages 553–572, 2010. References are to full version: `http://crypto.stanford.edu/~dabo/pubs/abstracts/latticebb.html`.
2. S. Agrawal, D. Boneh, and X. Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, pages 98–115, 2010.
3. M. Ajtai. Generating hard instances of the short basis problem. In *Automata, Languages and Programming, 26th International Colloquium, ICALP'99, Prague, Czech Republic, July 11-15, 1999, Proceedings*, pages 1–9, 1999.
4. J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. In *26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009, February 26-28, 2009, Freiburg, Germany, Proceedings*, pages 75–86, 2009.
5. B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. In *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, pages 1–18, 2001.

6. B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. *J. ACM*, 59(2):6, 2012.
7. M. Bellare and T. Ristenpart. Simulation without the artificial abort: Simplified proof and improved concrete security for waters' IBE scheme. In *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, pages 407–424, 2009.
8. D. Boneh, A. Sahai, and B. Waters. Functional encryption: Definitions and challenges. In *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings*, pages 253–273, 2011.
9. D. Boneh, G. Segev, and B. Waters. Targeted malleability: homomorphic encryption for restricted computations. In *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*, pages 350–366, 2012.
10. D. Boneh and B. Waters. Constrained pseudorandom functions and their applications. In *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part II*, pages 280–300, 2013.
11. D. Boneh and M. Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, pages 480–499, 2014.
12. E. Boyle, S. Goldwasser, and I. Ivan. Functional signatures and pseudorandom functions. In *Public-Key Cryptography - PKC 2014 - 17th International Conference on Practice and Theory in Public-Key Cryptography, Buenos Aires, Argentina, March 26-28, 2014. Proceedings*, pages 501–519, 2014.
13. Z. Brakerski, C. Gentry, and S. Halevi. Packed ciphertexts in lwe-based homomorphic encryption. In *Public Key Cryptography*, pages 1–13, 2013.
14. Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *ITCS*, pages 309–325, 2012.
15. Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. Classical hardness of learning with errors. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 575–584, 2013.
16. Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. In *FOCS*, pages 97–106, 2011.
17. Z. Brakerski and V. Vaikuntanathan. Fully homomorphic encryption from ring-lwe and security for key dependent messages. In *CRYPTO*, pages 505–524, 2011.
18. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, pages 207–222, 2004.
19. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*, pages 523–552, 2010.
20. M. Chase, M. Kohlweiss, A. Lysyanskaya, and S. Meiklejohn. Malleable proof systems and applications. In *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, pages 281–300, 2012.

21. J. H. Cheon, J.-S. Coron, J. Kim, M. S. Lee, T. Lepoint, M. Tibouchi, and A. Yun. Batch fully homomorphic encryption over the integers. In *EUROCRYPT*, pages 315–335, 2013.
22. R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, pages 45–64, 2002.
23. Y. Desmedt, V. Iovino, G. Persiano, and I. Visconti. Controlled homomorphic encryption: Definition and construction. *IACR Cryptology ePrint Archive*, 2014:989, 2014.
24. D. Dolev, C. Dwork, and M. Naor. Nonmalleable cryptography. *SIAM J. Comput.*, 30(2):391–437, 2000.
25. K. Emura, G. Hanaoka, G. Ohtake, T. Matsuda, and S. Yamada. Chosen ciphertext secure keyed-homomorphic public-key encryption. In *Public-Key Cryptography - PKC 2013 - 16th International Conference on Practice and Theory in Public-Key Cryptography, Nara, Japan, February 26 - March 1, 2013. Proceedings*, pages 32–50, 2013.
26. S. Garg, C. Gentry, S. Halevi, and M. Raykova. Two-round secure MPC from indistinguishability obfuscation. In *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014, San Diego, CA, USA, February 24-26, 2014. Proceedings*, pages 74–94, 2014.
27. S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 40–49, 2013.
28. C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178, 2009.
29. C. Gentry, S. Halevi, and N. P. Smart. Better bootstrapping in fully homomorphic encryption. In *Public Key Cryptography*, pages 1–16, 2012.
30. C. Gentry, S. Halevi, and N. P. Smart. Fully homomorphic encryption with polylog overhead. In *EUROCRYPT*, pages 465–482, 2012.
31. C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO (1)*, pages 75–92, 2013.
32. M. Green and G. Ateniese. Identity-based proxy re-encryption. In *Applied Cryptography and Network Security, 5th International Conference, ACNS 2007, Zhuhai, China, June 5-8, 2007, Proceedings*, pages 288–306, 2007.
33. D. Hofheinz, A. Kamath, V. Koppula, and B. Waters. Adaptively secure constrained pseudorandom functions. *IACR Cryptology ePrint Archive*, 2014:720, 2014.
34. D. Hofheinz and E. Kiltz. Programmable hash functions and their applications. In *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*, pages 21–38, 2008.
35. S. Hohenberger, V. Koppula, and B. Waters. Adaptively secure puncturable pseudorandom functions in the standard model. *IACR Cryptology ePrint Archive*, 2014:521, 2014.
36. S. Hohenberger, A. Sahai, and B. Waters. Replacing a random oracle: Full domain hash from indistinguishability obfuscation. In *Advances in Cryptology - EURO-*

CRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings, pages 201–220, 2014.

37. C. S. Jutla and A. Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I*, pages 1–20, 2013.
38. A. Kiayias, S. Papadopoulos, N. Triandopoulos, and T. Zacharias. Delegatable pseudorandom functions and applications. In *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013*, pages 669–684, 2013.
39. B. Libert, T. Peters, M. Joye, and M. Yung. Linearly homomorphic structure-preserving signatures and their applications. In *CRYPTO (2)*, pages 289–307, 2013.
40. B. Libert, T. Peters, M. Joye, and M. Yung. Non-malleability from malleability: Simulation-sound quasi-adaptive NIZK proofs and cca2-secure encryption from homomorphic signatures. In *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, pages 514–532, 2014.
41. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, pages 700–718, 2012.
42. M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC*, pages 427–437, 1990.
43. C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 333–342, 2009.
44. M. Prabhakaran and M. Rosulek. Rerandomizable RCCA encryption. In *Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings*, pages 517–534, 2007.
45. M. Prabhakaran and M. Rosulek. Homomorphic encryption with CCA security. In *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations*, pages 667–678, 2008.
46. M. Prabhakaran and M. Rosulek. Towards robust computation on encrypted data. In *Advances in Cryptology - ASIACRYPT 2008, 14th International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, Australia, December 7-11, 2008. Proceedings*, pages 216–233, 2008.
47. C. Rackoff and D. R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *CRYPTO*, pages 433–444, 1991.
48. K. Ramchen and B. Waters. Fully secure and fast signing from obfuscation. *IACR Cryptology ePrint Archive*, 2014:523, 2014.
49. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 84–93, 2005.
50. R. L. Rivest, L. Adleman, and M. L. Dertouzos. On data banks and privacy homomorphisms. *Foundations of secure computation*, 32(4):169–178, 1978.

51. A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA*, pages 543–553, 1999.
52. A. Sahai and B. Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 475–484, 2014.
53. N. P. Smart and F. Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In *Public Key Cryptography*, pages 420–443, 2010.
54. D. Stehlé and R. Steinfeld. Faster fully homomorphic encryption. In *ASIACRYPT*, pages 377–394, 2010.
55. M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In *EUROCRYPT*, pages 24–43, 2010.
56. B. Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT*, pages 114–127, 2005.
57. B. Waters. A punctured programming approach to adaptively secure functional encryption. *IACR Cryptology ePrint Archive*, 2014:588, 2014.