Chosen-Ciphertext Secure Multi-Identity and Multi-Attribute Pure FHE

Tapas Pal, Ratna Dutta

Department of Mathematics, Indian Institute of Technology Kharagpur
Kharagpur-721302, India
tapas.pal@iitkgp.ac.in,ratna@maths.iitkgp.ernet.in

Abstract. A *multi-identity pure fully homomorphic encryption* (MIFHE) enables a server to perform arbitrary computation on the ciphertexts that are encrypted under different identities. In case of *multi-attribute pure* FHE (MAFHE), the ciphertexts are associated with different attributes. Clear and McGoldrick (CANS 2014) gave the first chosen-plaintext attack secure MIFHE and MAFHE based on indistinguishability obfuscation. In this study, we focus on building MIFHE and MAFHE which are secure under type 1 of chosen-ciphertext attack (CCA1) security model. In particular, using witness pseudorandom functions (Zhandry, TCC 2016) and multi-key pure FHE or MFHE (Mukherjee and Wichs, EUROCRYPT 2016) we propose the following constructions:

- CCA secure *identity-based encryption* (IBE) that enjoys an *optimal* size ciphertexts, which we extend to a CCA1 secure MIFHE scheme.
- CCA secure *attribute-based encryption* (ABE) having an optimal size ciphertexts, which we transform into a CCA1 secure MAFHE scheme.

By optimal size, we mean that the bit-length of a ciphertext is the bit-length of the message plus a security parameter multiplied with a constant. Known constructions of multi-identity(attribute) FHEs are either leveled, that is, support only bounded depth circuit evaluations or secure in a weaker CPA security model. With our new approach, we achieve both CCA1 security and evaluation on arbitrary depth circuits for multi-identity(attribute) FHE schemes.

Keywords: witness pseudorandom function, identity-based encryption, attribute-based encryption, fully homomorphic encryption.

1 Introduction

Gentry settled the open problem of computing on encrypted data by proposing the first fully homomorphic encryption (FHE) [17] scheme based on ideal lattices. Afterwards, many researchers developed improved variants of Gentry's FHE [31,6,7]. These are all *leveled* FHE where a bounded depth circuit can be evaluated on encrypted data. While the error in an evaluated ciphertext may blow up with increasing depth, Gentry's bootstrapping technique [17] can be applied to convert any leveled FHE into a *pure* FHE which handles arbitrary depth circuits. The bootstrapping relies on circular security means that the scheme is secure even when the adversary is given an encryption of the secret-key.

Identity-based encryption (IBE) [3] gives us the freedom to encrypt data using any arbitrary string (treated as identity) instead of a specified public-key. Constructing identity-based FHE (IBFHE) remained difficult due to the presence of evaluation key until Gentry, Sahai and Waters [18] built a leveled FHE based on learning with errors (LWE) where the public parameters serve the role of the evaluation key. Compiling existing LWE-based identity-based encryption or LWE-based attribute-based encryption (ABE) with their FHE, [18] came up with efficient IBFHE and attribute-based FHE (ABFHE). Clear et al. [12] extends the IBFHE of [18] to multi-identity setting where evaluation can be performed with multiple users data and decryption requires a collaboration of their secret-keys. However, Gentry's bootstrapping theorem can not be applied to convert a leveled IBFHE (or ABFHE) into a pure IBFHE (or pure ABFHE). Since evaluation requires encryption of the secret-keys under the respective identities, the transformed IBFHE becomes interactive which is noted as *weak* [7].

To build a pure IBFHE, Clear and McGoldrick [11] used indistinguishability obfuscation ($i\mathcal{O}$) [30] and a pure FHE scheme. Specifically, they utilized the punctured technique of [30] to create a unique public-secret FHE key pair corresponding to an identity. The IBFHE can be extended to multi-identity pure FHE (MIFHE) when we use a multi-key pure FHE (MFHE) [27] in place of the normal FHE. The work [11] also described a multi-attribute pure FHE (MAFHE) using $i\mathcal{O}$. MAFHE enables us to encrypt messages under different attributes instead of users identities. A generic construction of (almost pure) MAFHE with a bounded number of parties was given in [10] which employs a MFHE and a leveled multi-attribute FHE.

All existing constructions of MIFHE or MAFHE [11,8] are either CPA secure or based on a powerful primitive $i\mathcal{O}$. In case of leveled variants of those primitives [18,12,5], known constructions have started from LWE-based IBE or ABE which mostly provide security in CPA model, hence the corresponding FHEs are inherently CPA secure. It is trivial to observe that CCA security can not be realized for FHE like primitives as evaluation is a public algorithm. But, we can still consider CCA1 security where the adversary is given access to the decryption oracle up-until it receives the challenge ciphertext. Canetti et al. [8] gave a generic construction of CCA1 secure MFHE from a CPA secure MIFHE and instantiated their (pure) MIFHE based on sub-exponential $i\mathcal{O}$. So we ask: *Can we build CCA1 secure MIFHE or MAFHE? Can we construct these primitives without using obfuscation?*

In this paper, we find out affirmative answers to those questions. Recently, Zhandry introduced a different type of pseudorandom function (PRF), called witness PRF (WPRF) [33], which can produce a pseudorandom value $y = \mathsf{F}(\mathsf{fk}, x)$ corresponding to an NP statement x using a secret function key fk and anyone holding a valid witness of x can recompute y using a public evaluation key ek. If a statement x does not belong to the NP language then y becomes indistinguishable from random. The primitive finds many applications in building cryptographic tools such as non-interactive multiparty key exchange, witness encryption (WE), poly-many hardcore bits for one-way functions (OWFs) [33] that are previously possible only from $i\mathcal{O}$. We aim to construct CCA1 secure MIFHE and MAFHE schemes using WPRF.

Zhandry [33] built WPRF from multilinear subset-sum Diffie-Hellman assumption which is a *target-group* assumption and hence most of the existing (source-group based) attacks on multilinear maps may not be a threat to the WPRF. On the other side, WPRF construction of [28] based on sublinear compact randomized encoding and puncturable PRF indicates that it belongs to obfustopia. However, WPRF is not known to imply $i\mathcal{O}$ and seems to be a much weaker assumption than $i\mathcal{O}$ [33]. Few primitives like *smooth projective hash functions* [13], *functional* PRFs [4] and *publicly evaluable* PRFs [9] that are close to the notion of WPRF have already been realized from standard assumptions. Therefore, it is more likely to realize WPRF from standard assumptions much before the community arrive at a practical construction of $i\mathcal{O}$.

Contribution. This work investigates applications of WPRF in identity-based and attribute-based cryptography.

1. Multi-Identity Pure FHE: In the era of cloud computing, it is highly desirable to run arbitrarily complex programs over any type of encrypted data. To compute on the ciphertexts of an IBE scheme, we build the first CCA1 secure MIFHE using WPRF and MFHE. The stepping-stone of our MIFHE is a CCA secure IBE that we construct from WPRF and a special signature scheme.

Our goal is to use OWFs along with WPRF to get a CCA secure IBE with short secret-keys and optimal size ciphertexts. In particular, we take a pseudorandom generator (PRG) and a secure signature scheme both of which can be efficiently realized from OWFs [29]. First we generate a pair of WPRF keys (fk, ek) for an NP language $L = \{(\mathsf{id}, v, \mathsf{vk}) : (\exists u \text{ such that } \mathsf{PRG}(\mathsf{id} \oplus u) = v) \text{ or } (\exists \sigma \text{ such that } \mathsf{Vrfy}(\mathsf{vk}, \mathsf{id}, \sigma) = 1)\}$ with a relation R where id is an identity and vk is a verification key of the signature scheme. The public-key of the IBE is a tuple (ek, vk) and the master secret-key is the signing key sk. A secret-key for an identity id is as short as a signature σ of id. At the time of encryption, we use ek to generate a pseudorandom value y corresponding to a statement $(\mathsf{id}, v, \mathsf{vk})$ with a witness u such that $\mathsf{PRG}(\mathsf{id} \oplus u) = v$. The ciphertext is a tuple (c_s, v) where c_s is a symmetric-key encryption (SKE) of a message m using y. Interestingly, the size of the ciphertext becomes optimal, that is $|m| + c\lambda$ where λ is a security parameter and c is a constant.

We need extractibility property of WPRF [33] to prove the security of IBE. However, we show (in Sec. 3) that the strong extractibility assumption can be avoided by replacing the normal signature scheme with a primitive called all-but-one signature (ABOS) [20]. We note that ABOS can be constructed from a verifiable random function (VRF) [26] and a perfectly-binding commitment scheme. Existing constructions [16,25] of CCA secure IBE achieve (almost) optimal ciphertexts based on bilinear maps. Our result shows that assuming VRF and a normal WPRF we can achieve a CCA secure IBE with optimal size ciphertexts. However, optimal ciphertext for IBE is not a primary contribution of this paper, rather we utilize our IBE to achieve more advanced primitive.

To convert the IBE into a MIFHE scheme (Sec. 3.1), we replace the SKE by a multi-key pure FHE which has been constructed using LWE assumption along with circular security [27]. In the pure MIFHE of [11] (based on obfuscation), the public-key of the underlying MFHE is unique for each identity, whereas there may be exponentially many MFHE public-keys associated to a single identity in our MIFHE and we have to include the MFHE public-key into a ciphertext so that evaluation runs smoothly. Therefore, MFHE is necessary for our construction even when messages are encrypted under the same identity.

2. Multi-Attribute Pure FHE: To achieve a $\mathsf{CCA1}$ secure MAFHE, we first realize a CCA secure attribute-based encryption (ABE) [32] using WPRF. Recall that a (key-policy) ABE enables us to encrypt messages under a set of attributes mapped to a bit-string x and a receiver holding a secret-key sk_f corresponding to a boolean function f should succeed in decrypting the ciphertext when x satisfies f. If we consider a WPRF for the language $L = \{(x, v, \mathsf{vk}) : (\exists u \text{ such that } \mathsf{PRG}(x \oplus u) = v) \text{ or } (\exists \sigma \text{ such that } \mathsf{Vrfy}(\mathsf{vk}, f, \sigma) = 1 \wedge f(x) = 1)\}$ similar to our basic IBE construction, then we can achieve a CCA secure ABE from OWFs. Here also we need to rely on extractability property of WPRF. To avoid this strong assumption, we start with the WE-based ABE of Garg et al. [15]. Specifically, the signature scheme is replaced with a witness-indistinguishable non-interactive zap [22]. The main difference from [15] is that to embed an attribute into a ciphertext we imitate the technique of embedding an identity from our IBE construction.

Goyal et al. [21] gave the first CCA secure ABE using bilinear maps. They used the generic technique of [2] to establish a bridge from CPA to CCA security for ABE. However, their transformation works in an environment where the CPA secure ABE has to support delegatability [21]. Another generic transformation was proposed in [32] which needs *verifiability* of a ciphertext encrypted under two different attributes. Our approach (in Sec. 4) defines a way to achieve a CCA secure ABE which is the first to enjoy an optimal ciphertext size (to the best of our knowledge).

We transform our ABE to a $\mathsf{CCA1}$ secure MAFHE scheme (in Sec. 4.1) following the similar technique employed in the conversion of our MIFHE from the IBE. The MIFHEs and MAFHEs of [11,10] are secure under the chosen-plaintext model which is often insufficient in many practical scenarios. Our approach leads to the first $\mathsf{CCA1}$ secure MIFHE and $\mathsf{CCA1}$ secure MAFHE without assuming $i\mathcal{O}$.

Other Related Works. Garg et al. [15] proposed constructions of IBE and ABE from witness encryption (WE) (introduced in the same work). Their selectively secure IBE is based on a dual encryption methodology and unique signature scheme. Replacing WE by WPRF does not immediately produce an optimal size ciphertext for the IBE. Using non-interactive zap and commitment schemes they built adaptively secure IBE and selectively secure ABE schemes. However, security holds in the CPA model and extension to MIFHE or MAFHE may require additional primitive like obfuscation. Goldwasser et al. [19] built an ABE for

Turing machines from WE and succinct argument of knowledge. But, their ABE is only CPA secure and based on strong extractibility assumptions.

2 Preliminaries

Notations. For any set S, the notation $x \leftarrow S$ denotes the process of sampling x uniformly at random from S. Let $\mathcal{E}$ be a probabilistic polynomial time (PPT) algorithm. Then $y \leftarrow \mathcal{E}(x)$ denotes the execution of $\mathcal{E}$ with an input x using fresh randomness and assign the output to y. If the randomness, say r, is provided externally then we denote this execution by $y \leftarrow \mathcal{E}(x; r)$. If $x \in \{0,1\}^*$ then we denote by $|x|$ the size of x. We say $f : \mathbb{N} \to \mathbb{R}$ is a *negligible* function of n if it is $O(n^{-c})$ for all $c > 0$, and we use $\mathsf{negl}(n)$ to denote a negligible function of n.

2.1 Pseudorandom Generator [1]

Definition 1 A pseudorandom generator (PRG) is a deterministic polynomial time algorithm PRG that on input a seed $s \in \{0,1\}^\lambda$ outputs a string of length $\ell(\lambda)$ such that the following holds:
- *expansion*: For every λ it holds that $\ell(\lambda) > \lambda$.
- *pseudorandomness*: For all PPT adversary $\mathcal{A}$ and $s \leftarrow \{0,1\}^\lambda, r \leftarrow \{0,1\}^{\ell(\lambda)}$, there exists a negligible function negl such that

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{PRG}}(\lambda) = |\Pr[\mathcal{A}(1^\lambda, \mathsf{PRG}(s)) = 1] - \Pr[\mathcal{A}(1^\lambda, r) = 1]\ | < \mathsf{negl}(\lambda).$$

2.2 Symmetric Key Encryption [23,24]

Definition 2 A symmetric key encryption (SKE) scheme is a tuple of PPT algorithms (Gen, Enc, Dec) defined as follows:
- $\mathsf{K} \leftarrow \mathsf{Gen}(1^\lambda)$: on input a security parameter λ, returns a key K.
- $c \leftarrow \mathsf{Enc}(\mathsf{K}, m)$: a randomized algorithm that returns c, an encryption of the message $m \in \mathcal{M}$.
- $\mathsf{Dec}(\mathsf{K}, c) \in \mathcal{M} \cup \{\perp\}$: a deterministic algorithm that decrypts the ciphertext c and returns a message $m \in \mathcal{M}$, or $\perp$ if it fails.

The SKE is said to be correct if the following holds:
- *correctness*: For all $m \in \mathcal{M}$ and $\mathsf{K} \leftarrow \mathsf{Gen}(1^\lambda)$, we require that

$$\Pr[\mathsf{Dec}(\mathsf{K}, \mathsf{Enc}(\mathsf{K}, m)) = m] = 1$$

We consider chosen ciphertext attack (CCA) security for SKE and define an experiment $\mathsf{Expt}_{\mathcal{A},\mathsf{CCA}}^{\mathsf{SKE}}(1^\lambda)$ in Fig. 1.

Definition 3 A symmetric key encryption SKE is said to satisfy chosen ciphertext attack (CCA) security if, for all PPT adversary $\mathcal{A}$, there exists a negligible function negl such that

$$\mathsf{Adv}_{\mathcal{A},\mathsf{CCA}}^{\mathsf{SKE}}(\lambda) = |\Pr[\mathsf{Expt}_{\mathcal{A},\mathsf{CCA}}^{\mathsf{SKE}}(1^\lambda) = 1] - \tfrac{1}{2}| < \mathsf{negl}(\lambda)$$

$\mathsf{K} \leftarrow \mathsf{Gen}(1^\lambda)$
$(m_0, m_1) \leftarrow \mathcal{A}^{\mathsf{Enc}(\mathsf{K},\cdot),\mathsf{Dec}(\mathsf{K},\cdot)}(1^\lambda)$
$b \leftarrow \{0,1\}$
$c \leftarrow \mathsf{Enc}(\mathsf{K}, m_b)$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}(\mathsf{K},\cdot),\mathsf{Dec}(\mathsf{K},\cdot)}(c)$
if $(b' = b) \wedge (c \notin Q_{\mathsf{K}})$
  return 1
Q_{K} = set of all $\mathsf{Dec}(\mathsf{K},\cdot)$ queries

Fig. 1: $\mathsf{Expt}^{\mathsf{SKE}}_{\mathcal{A},\mathsf{CCA}}(1^\lambda)$

$m^* \leftarrow \mathcal{A}(1^\lambda)$
$(\mathsf{sk}_0, \mathsf{vk}_0) \leftarrow \mathsf{Setup}(1^\lambda)$
$(\mathsf{sk}_1, \mathsf{vk}_1) \leftarrow \mathsf{PuncSetup}(1^\lambda, m^*)$
$b \leftarrow \{0,1\}$
$b' \leftarrow \mathcal{A}^{\mathsf{Sig}(\mathsf{sk}_b,\cdot)}(\mathsf{vk}_b)$
if $(b = b') \wedge (m^* \notin Q_{\mathsf{sk}})$
  return 1
Q_{sk} = set of all $\mathsf{Sig}(\mathsf{sk}_b,\cdot)$ queries

Fig. 2: $\mathsf{Expt}^{\mathsf{ABOS}}_{\mathcal{A}}(1^\lambda)$

$x^* \leftarrow \mathcal{A}(1^\lambda)$
$(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{Gen}(1^\lambda, R)$
$y_0 \leftarrow \mathsf{F}(\mathsf{fk}, x^*)$, $y_1 \leftarrow \mathcal{Y}$
$b \leftarrow \{0,1\}$
$b' \leftarrow \mathcal{A}^{\mathsf{F}(\mathsf{fk},\cdot)}(\mathsf{ek}, y_b)$
if $(b' = b) \wedge (x^* \notin L \cup Q_{\mathsf{fk}})$
  return 1
Q_{fk} = set of all $\mathsf{F}(\mathsf{fk},\cdot)$ queries

Fig. 3: $\mathsf{Expt}^{\mathsf{WPRF},R}_{\mathcal{A}}(1^\lambda)$

Remark 1 We take a *length preserving* SKE means $|\mathsf{Enc}(\mathsf{K}, m)| = |m|$. In such a scheme, $\mathcal{A}$ is not allowed to query m_0 and m_1 to the encryption oracle. The CMC mode [23] and ECM mode [24], proposed by Halevi and Rogaway, is length preserving and CCA secure if the underlying block cipher is a strong pseudorandom permutation such as AES [14]. In fact, we need much weaker notion of CCA security where $\mathcal{A}$ is not given the access of $\mathsf{Enc}(\mathsf{K},\cdot)$. We term this notion as length preserving CCA (LP-CCA) secure SKE which is sufficient for our applications.

2.3 All-but-one Signature Scheme [20]

Definition 4 An all-but-one signature (ABOS) scheme is a tuple of PPT algorithms (Setup, PuncSetup, Sig, Vrfy) defined as follows:
- $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Setup}(1^\lambda)$: on input a security parameter λ, outputs a signing key sk and a verification key vk.
- $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{PuncSetup}(1^\lambda, m^*)$: on input a security parameter λ and a message $m^* \in \mathcal{M}$, outputs a signing key sk and a verification key vk.
- $\sigma \leftarrow \mathsf{Sig}(\mathsf{sk}, m)$: returns $\sigma \in \Sigma$, a signature of the message $m \in \mathcal{M}$.
- $\mathsf{Vrfy}(\mathsf{vk}, m, \sigma) \in \{0,1\}$: a deterministic algorithm that on input a verification key vk, a message m and a signature σ, and outputs either 0 or 1.

The signature scheme ABOS is said to be correct if the following holds:
– *correctness of* Setup: For all $m \in \mathcal{M}$ and $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Setup}(1^\lambda)$, we require

$$\Pr[\mathsf{Vrfy}(\mathsf{vk}, m, \mathsf{Sig}(\mathsf{sk}, m)) = 1] = 1$$

– *correctness of* PuncSetup: For any $m^* \in \mathcal{M}$, $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{PuncSetup}(1^\lambda, m^*)$ and any $\sigma \in \Sigma$, we have $\mathsf{Vrfy}(\mathsf{vk}, m^*, \sigma) = 0$.

We consider VK indistinguishability experiment $\mathsf{Expt}^{\mathsf{ABOS}}_{\mathcal{A}}(1^\lambda)$ in Fig. 2.

Definition 5 An all-but-one signature ABOS scheme is said to satisfy VK indistinguishability (VK-IND) security if for all PPT adversary $\mathcal{A}$, there exists a negligible function negl such that

$$\mathsf{Adv}^{\mathsf{ABOS}}_{\mathcal{A}}(\lambda) = |\Pr[\mathsf{Expt}^{\mathsf{ABOS}}_{\mathcal{A}}(1^\lambda) = 1] - \tfrac{1}{2}| < \mathsf{negl}(\lambda)$$

1. $\mathsf{id}^* \leftarrow \mathcal{A}(1^\lambda)$
2. $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$
3. $(m_0, m_1) \leftarrow \mathcal{A}^{O_{\mathsf{sk}}(\cdot), O_{\mathsf{D}}(\cdot)}(\mathsf{pp})$
4. $b \leftarrow \{0, 1\}$
5. $c^* \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{id}^*, m_b)$
6. $b' \leftarrow \mathcal{A}^{O_{\mathsf{sk}}(\cdot), O_{\mathsf{D}}(\cdot)}(c^*)$
7. return 1 if $(b' = b) \wedge (|m_0| = |m_1|)$

$\underline{O_{\mathsf{sk}}(\cdot)}$:
1. input: $\mathsf{id} \in \mathcal{ID}$
2. compute $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{id})$
3. return $\mathsf{sk}_{\mathsf{id}}$ if $\mathsf{id} \neq \mathsf{id}^*$, else $\perp$

$\underline{O_{\mathsf{D}}(\cdot)}$:
1. input: $(\mathsf{id} \in \mathcal{ID},\ c)$
2. compute $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{id})$
3. return $\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_{\mathsf{id}}, c)$
   if $(\mathsf{id}, c) \neq (\mathsf{id}^*, c^*)$, else $\perp$

Fig. 4: $\mathsf{Expt}^{\mathsf{IBE}}_{\mathcal{A},\mathsf{CCA}}(1^\lambda)$

2.4 Witness Pseudorandom Function [33]

Definition 6 A witness pseudorandom function (WPRF) for an NP language L with a relation R is a tuple of PPT algorithms (Gen, F, Eval) defined as follows:

- $(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{Gen}(1^\lambda, R)$: on input a security parameter λ and a relation circuit $R : \mathcal{X} \times \mathcal{W} \rightarrow \{0, 1\}$, returns a secret function key fk and a public evaluation key ek.
- $y \leftarrow \mathsf{F}(\mathsf{fk}, x)$: returns a pseudorandom value $y \in \mathcal{Y}$ for $x \in \mathcal{X}$.
- $\mathsf{Eval}(\mathsf{ek}, x, w) \in \mathcal{Y} \cup \{\perp\}$: on input an evaluation key ek, an element $x \in \mathcal{X}$ and a witness $w \in \mathcal{W}$, returns an element $y \in \mathcal{Y}$, or $\perp$ if it fails.

We note that, each of the above algorithms except Gen is a deterministic algorithm. The WPRF is said to be correct if the following holds:

– *correctness of* Eval: For all $x \in \mathcal{X}, w \in \mathcal{W}$ and $(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{Gen}(1^\lambda, R)$, we require that

$$\mathsf{Eval}(\mathsf{ek}, x, w) = \begin{cases} \mathsf{F}(\mathsf{fk}, x) & \text{if } R(x, w) = 1 \\ \perp & \text{if } R(x, w) = 0 \end{cases}$$

The security experiment $\mathsf{Expt}^{\mathsf{WPRF},R}_{\mathcal{A}}(1^\lambda)$ for the WPRF is defined in Fig. 3. We consider a selective model which is sufficient for our applications.

Definition 7 A witness pseudorandom function WPRF for an NP language L with a relation R is said to be selectively secure if, for all PPT adversary $\mathcal{A}$, there exists a negligible function negl such that

$$\mathsf{Adv}^{\mathsf{WPRF},R}_{\mathcal{A}}(\lambda) = |\Pr[\mathsf{Expt}^{\mathsf{WPRF},R}_{\mathcal{A}}(1^\lambda) = 1] - \tfrac{1}{2}| < \mathsf{negl}(\lambda)$$

3 CCA1 Secure MIFHE from WPRF and MFHE

The main building block of our MIFHE is a CCA secure IBE. Firstly, we use WPRF and ABOS to achieve a CCA secure IBE having an optimal size ciphertext. Then we extend it to a CCA1 secure MIFHE with the help of existing MFHE schemes. We begin with the definition of an IBE system.

Definition 8 [3] An identity-based encryption (IBE) scheme is a tuple of PPT algorithms (Setup, KeyGen, Enc, Dec) defined as follows:

- $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$: on input a security parameter λ, produces a public parameter pp and a master secret-key msk.
- $\mathsf{sk_{id}} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{id})$: returns a secret-key $\mathsf{sk_{id}}$ corresponding to the identity $\mathsf{id} \in \mathcal{ID}$ using a master secret-key msk.
- $c \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{id}, m)$: returns c, an encryption of a message $m \in \mathcal{M}$ under an identity id.
- $\mathsf{Dec}(\mathsf{pp}, \mathsf{sk_{id}}, c) \in \mathcal{M} \cup \{\perp\}$: a deterministic algorithm that decrypts a ciphertext c using a secret-key $\mathsf{sk_{id}}$ and outputs either a message $m \in \mathcal{M}$ or $\perp$ if it fails.

The IBE is said to be correct if the following holds:
- *correctness*: For all $\mathsf{id} \in \mathcal{ID}$, $m \in \mathcal{M}$, $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and $\mathsf{sk_{id}} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{id})$, we require that

$$\Pr[\mathsf{Dec}(\mathsf{pp}, \mathsf{sk_{id}}, \mathsf{Enc}(\mathsf{pp}, \mathsf{id}, m)) = m] = 1$$

For security of IBE, we consider CCA security with selective-identity experiment $\mathsf{Expt}^{\mathsf{IBE}}_{\mathcal{A},\mathsf{CCA}}(1^\lambda)$ described in Fig. 4.

Definition 9 An identity-based encryption IBE is said to be selective-identity CCA secure if, for all PPT adversary $\mathcal{A}$, there exists a negligible function negl such that

$$\mathsf{Adv}^{\mathsf{IBE}}_{\mathcal{A},\mathsf{CCA}}(\lambda) = |\Pr[\mathsf{Expt}^{\mathsf{IBE}}_{\mathcal{A},\mathsf{CCA}}(1^\lambda) = 1] - \tfrac{1}{2}| < \mathsf{negl}(\lambda)$$

Construction. We construct an identity-based encryption scheme IBE = (Setup, KeyGen, Enc, Dec) for an identity space $\mathcal{ID} = \{0,1\}^\lambda$. The following primitives are utilized:
- A pseudorandom generator $\mathsf{PRG} : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$.
- A $\mathsf{LP\text{-}CCA}$ secure symmetric key encryption SKE = (Gen, Enc, Dec).
- A $\mathsf{VK\text{-}IND}$ secure all-but-one signature scheme ABOS = (Setup, $\mathsf{PuncSetup}$, Sig, Vrfy) with the message space as $\mathcal{ID}$ and signature space Σ.
- A WPRF = (Gen, F, Eval) for the NP language $L = \{(\mathsf{id}, v, \mathsf{vk}) : (\exists u \in \{0,1\}^\lambda$ such that $\mathsf{PRG}(\mathsf{id} \oplus u) = v)$ or $(\exists \sigma$ such that $\mathsf{ABOS.Vrfy}(\mathsf{vk}, \mathsf{id}, \sigma) = 1)\}$ with a relation $R : \mathcal{X} \times \mathcal{W} \to \{0,1\}$. So, $R((\mathsf{id}, v, \mathsf{vk}), \omega) = 1$ if $(\mathsf{PRG}(\mathsf{id} \oplus \omega) = v) \vee (\mathsf{Vrfy}(\mathsf{vk}, \mathsf{id}, \omega) = 1)$, 0 otherwise. Note that, we can always fix the input size of R by adding some dummy bits.

We describe our IBE in Fig. 5. For *correctness*, we have to make sure that a same pseudorandom value y is generated in both the algorithms Enc and Dec. In Enc, we compute y using a witness u for PRG and in Dec, we compute y using a witness which is now a signature σ for id. More importantly, the statement $(\mathsf{id}, v, \mathsf{vk})$ remains unchanged in both cases. Thus, correctness of Eval ensures $y = \mathsf{WPRF.F}(\mathsf{fk}, (\mathsf{id}, v, \mathsf{vk}))$ is the same in Enc and Dec. Finally, Dec returns the message m using the decryption of SKE.

Efficiency: The ciphertext size of our IBE is compact in the sense that it has only $|c_s| + |v|$ many bits. Since c_s is a ciphertext of a length preserving SKE, we have $|c_s| = |m|$, where $|m|$ denotes the bit length of message. Therefore, the size

$\underline{\mathsf{Setup}(1^\lambda)}$:
1. $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{ABOS.Setup}(1^\lambda)$
2. $(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{WPRF.Gen}(1^\lambda, R)$
3. set $\mathsf{pp} = (\mathsf{ek}, \mathsf{vk})$, $\mathsf{msk} = \mathsf{sk}$
4. return $(\mathsf{pp}, \mathsf{msk})$

$\underline{\mathsf{KeyGen}(\mathsf{msk}, \mathsf{id})}$:
1. parse $\mathsf{msk} = \mathsf{sk}$
2. $\sigma \leftarrow \mathsf{ABOS.Sig}(\mathsf{sk}, \mathsf{id})$
3. set $\mathsf{sk}_{\mathsf{id}} = (\sigma, \mathsf{id})$
4. return $\mathsf{sk}_{\mathsf{id}}$

$\underline{\mathsf{Enc}(\mathsf{pp}, \mathsf{id}, m)}$:
1. parse $\mathsf{pp} = (\mathsf{ek}, \mathsf{vk})$
2. $u \leftarrow \{0,1\}^\lambda$, $v \leftarrow \mathsf{PRG}(\mathsf{id} \oplus u)$
3. $y \leftarrow \mathsf{WPRF.Eval}(\mathsf{ek}, (\mathsf{id}, v, \mathsf{vk}), u)$
4. $\mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y)$
5. $c_s \leftarrow \mathsf{SKE.Enc}(\mathsf{K}, m)$
6. return $c = (c_s, v)$

$\underline{\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_{\mathsf{id}}, c)}$:
1. parse $\mathsf{pp} = (\mathsf{ek}, \mathsf{vk})$
2. parse $\mathsf{sk}_{\mathsf{id}} = (\sigma, \mathsf{id})$, $c = (c_s, v)$
3. $y \leftarrow \mathsf{WPRF.Eval}(\mathsf{ek}, (\mathsf{id}, v, \mathsf{vk}), \sigma)$
4. $\mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y)$
5. return $\mathsf{SKE.Dec}(\mathsf{K}, c_s)$

Fig. 5: Construction of IBE with optimal ciphertexts

of c is $|m| + 2\lambda$ which is *optimal* for any IBE scheme. The underlying relation R is also simple as it either checks a PRG or verify a message-signature pair. This means the size of public parameter is proportional to the size of PRG plus the size of Vrfy, hence is some fixed polynomial in λ.

Theorem 1 *The* $\mathsf{IBE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *described above is a selective-identity* CCA *secure identity based encryption if* PRG *is a secure pseudorandom generator,* WPRF *is a selectively secure witness pseudorandom function,* ABOS *is a* VK-IND *secure all-but-one signature scheme and* SKE *is a* LP-CCA *secure symmetric key encryption.*

Proof. We prove the security of IBE using the following sequence of games. As usual, we start with Game 0 which is the standard experiment $\mathsf{Expt}_{\mathcal{A}}^{\mathsf{IBE}}(\lambda)$ as defined in Fig. 4. For Game i, let $\mathsf{G_i}$ be the event $b = b'$. We assume that $\mathcal{A}$ submits two messages of equal length in each game.

<u>Game 0:</u> This is the standard experiment as described in Def. 9. In particular, $\mathcal{A}$ begins by committing to a challenge identity id^*. The challenger computes $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and transfers pp to $\mathcal{A}$. The adversary, given access to the oracles $O_{\mathsf{sk}}(\cdot), O_{\mathsf{D}}(\cdot)$, submits a pair of challenge messages (m_0, m_1). Next, the challenger chooses a random bit b and sends the challenge ciphertext as $c^* \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{id}^*, m_b)$. Finally, $\mathcal{A}$, given access to the same oracles, guesses the challenge bit b. Note that, $\mathcal{A}$ cannot make a query id^* to $O_{\mathsf{sk}}(\cdot)$ and a query (id^*, c^*) to $O_{\mathsf{D}}(\cdot)$.

<u>Game 1:</u> It is same as Game 0 except that the challenger generates the randomness as $y \leftarrow \mathsf{WPRF.F}(\mathsf{fk}, (\mathsf{id}^*, v, \mathsf{vk}))$ instead of using Eval with the witness u. Game 1 is described in Fig. 6. It can be observed by the correctness of Eval

$$\mathsf{WPRF.Eval}(\mathsf{ek}, (\mathsf{id}^*, v, \mathsf{vk}), u) = \mathsf{WPRF.F}(\mathsf{fk}, (\mathsf{id}^*, v, \mathsf{vk}))$$

as $R((\mathsf{id}^*, v, \mathsf{vk}), u) = 1$. Therefore, the ciphertext distributions in games 0 and 1 are identical. This implies $\Pr[\mathsf{G_0}] = \Pr[\mathsf{G_1}]$.

1. $\mathsf{id}^* \leftarrow \mathcal{A}(1^\lambda)$
2. $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{ABOS.Setup}(1^\lambda)$
3. $(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{WPRF.Gen}(1^\lambda, R)$
4. set $\mathsf{pp} = (\mathsf{ek}, \mathsf{vk})$, $\mathsf{msk} = \mathsf{sk}$
5. $(m_0, m_1) \leftarrow \mathcal{A}^{O_{\mathsf{sk}}(\cdot), O_{\mathsf{D}}(\cdot)}(\mathsf{pp})$
6. $u \leftarrow \{0,1\}^\lambda$, $v \leftarrow \mathsf{PRG}(\mathsf{id}^* \oplus u)$
7. $\boxed{y \leftarrow \mathsf{WPRF.F}(\mathsf{fk}, (\mathsf{id}^*, v, \mathsf{vk}))}$
8. $\mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y)$
9. $b \leftarrow \{0,1\}$
10. $c_s^* \leftarrow \mathsf{SKE.Enc}(\mathsf{K}, m_b)$
11. set $c^* = (c_s^*, v)$
12. $b' \leftarrow \mathcal{A}^{O_{\mathsf{sk}}(\cdot), O_{\mathsf{D}}(\cdot)}(c^*)$
13. return 1 if $(b' = b)$

Fig. 6: Game 1

1. $\mathsf{id}^* \leftarrow \mathcal{A}(1^\lambda)$
2. $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{ABOS.Setup}(1^\lambda)$
3. $(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{WPRF.Gen}(1^\lambda, R)$
4. set $\mathsf{pp} = (\mathsf{ek}, \mathsf{vk})$, $\mathsf{msk} = \mathsf{sk}$
5. $(m_0, m_1) \leftarrow \mathcal{A}^{O_{\mathsf{sk}}(\cdot), O_{\mathsf{D}}(\cdot)}(\mathsf{pp})$
6. $\boxed{v \leftarrow \{0,1\}^{2\lambda}}$
7. $y \leftarrow \mathsf{WPRF.F}(\mathsf{fk}, (\mathsf{id}^*, v, \mathsf{vk}))$
8. $\mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y)$
9. $b \leftarrow \{0,1\}$
10. $c_s^* \leftarrow \mathsf{SKE.Enc}(\mathsf{K}, m_b)$
11. set $c^* = (c_s^*, v)$
12. $b' \leftarrow \mathcal{A}^{O_{\mathsf{sk}}(\cdot), O_{\mathsf{D}}(\cdot)}(c^*)$
13. return 1 if $(b' = b)$

Fig. 7: Game 2

Game 2: It is exactly same as Game 1 except that the challenger picks v uniformly at random from $\{0,1\}^{2\lambda}$ instead of computing $v \leftarrow \mathsf{PRG}(\mathsf{id}^* \oplus u)$. Game 2 is described in Fig. 7. Since u is chosen uniformly at random from $\{0,1\}^\lambda$, the distribution of $\mathsf{id}^* \oplus u$ is also uniform over $\{0,1\}^\lambda$. The security of PRG (Def. 1) ensures that $\mathcal{A}$'s advantage in distinguishing between Game 1 and Game 2 is $|\Pr[\mathsf{G}_1] - \Pr[\mathsf{G}_2]| = \mathsf{Adv}^{\mathsf{PRG}}_{\mathcal{B}_1}(\lambda) = \mathsf{negl}(\lambda)$ where $\mathcal{B}_1$ is a PRG-adversary.

Game 3: It is similar to Game 2 except that the challenger computes $(\mathsf{sk}^*, \mathsf{vk}^*) \leftarrow \mathsf{ABOS.PuncSetup}(1^\lambda, \mathsf{id}^*)$ in the setup and replaces the key generation and decryption oracles with $O_{\mathsf{sk}^*}(\cdot)$ and $O_{\mathsf{D},\mathsf{vk}^*,\mathsf{K}}(\cdot)$ respectively, defined in Fig. 8. Therefore, $\mathcal{A}$ gets a public parameter of the form $\mathsf{pp} = (\mathsf{ek}, \mathsf{vk}^*)$. In Lemma 1, we show that Game 2 and Game 3 are indistinguishable from $\mathcal{A}$'s view.

Game 4: It is identical to Game 3 except that the challenger selects y uniformly at random from $\mathcal{Y}$ which is the co-domain of $\mathsf{WPRF.F}(\mathsf{fk}, \cdot)$ and replaces the decryption oracle $O_{\mathsf{D},\mathsf{vk}^*,\mathsf{K}}(\cdot)$ by $O_{\mathsf{D}^*,\mathsf{vk}^*,\mathsf{K}}(\cdot)$, defined in Fig. 9. In Lemma 2, we show that Game 3 and Game 4 are indistinguishable from $\mathcal{A}$'s view.

Finally, we note that the encryption key in Game 4 is computed as $\mathsf{K} \leftarrow \mathsf{SKE.Gen}$
$(1^\lambda; y)$ where y is a fresh randomness which is independent of the challenge identity id^*. Hence, by the LP-CCA security of SKE (Remark 1) we have $|\Pr[\mathsf{G}_4] - \frac{1}{2}| = \mathsf{Adv}^{\mathsf{SKE}}_{\mathcal{B}_2,\mathsf{LP\text{-}CCA}}(\lambda)$ which is negligible in λ by our assumption. We are left to prove the following lemmas to conclude the security of our IBE.

Lemma 1 *Assuming* ABOS *is a* VK-IND *secure all-but-one signature scheme, we have* $|\Pr[\mathsf{G}_2] - \Pr[\mathsf{G}_3]| = \mathsf{negl}(\lambda)$.

Proof. We show that if $\mathcal{A}$ can distinguish between the games 2 and 3, then there exists an adversary $\mathcal{B}_3$ which will break the VK-IND security of ABOS (Def. 5). Let id^* be the challenge message for $\mathcal{B}_3$ which simulates $\mathcal{A}$ as follows:
$\mathcal{B}_3(1^\lambda, \mathsf{id}^*)$:
1. send id^* to its challenger

$\mathsf{id}^* \leftarrow \mathcal{A}(1^\lambda)$
$\boxed{(\mathsf{sk}^*, \mathsf{vk}^*) \leftarrow \mathsf{ABOS.PuncSetup}(1^\lambda, \mathsf{id}^*)}$
$(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{WPRF.Gen}(1^\lambda, R)$
set $\mathsf{pp} = (\mathsf{ek}, \mathsf{vk}^*)$ and $\mathsf{msk} = \mathsf{sk}^*$
$v \leftarrow \{0,1\}^{2\lambda}$
$y^* \leftarrow \mathsf{WPRF.F}(\mathsf{fk}, (\mathsf{id}^*, v, \mathsf{vk}^*))$
$\mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y^*)$
$(m_0, m_1) \leftarrow \mathcal{A}^{O_{\mathsf{sk}^*}(\cdot), O_{\mathsf{D},\mathsf{vk}^*,\mathsf{K}}(\cdot)}(\mathsf{pp})$
$b \leftarrow \{0,1\}$
$c_s^* \leftarrow \mathsf{SKE.Enc}(\mathsf{K}, m_b)$
return $c^* = (c_s^*, v)$
$b' \leftarrow \mathcal{A}^{O_{\mathsf{sk}^*}(\cdot), O_{\mathsf{D},\mathsf{vk}^*,\mathsf{K}}(\cdot)}(c^*)$
return 1 if $(b' = b)$

$\underline{O_{\mathsf{sk}^*}(\cdot)}$:
input: $\mathsf{id} \in \mathcal{ID}$
compute $\sigma \leftarrow \mathsf{ABOS.Sig}(\mathsf{sk}^*, \mathsf{id})$
return $\mathsf{sk}_{\mathsf{id}} = (\mathsf{id}, \sigma)$ if $\mathsf{id} \neq \mathsf{id}^*$, else $\perp$

$\underline{O_{\mathsf{D},\mathsf{vk}^*,\mathsf{K}}(\cdot)}$:
input: $(\mathsf{id} \in \mathcal{ID}, c)$
parse $c = (\bar{c}_s, \bar{v})$
if $(\mathsf{id}, c) = (\mathsf{id}^*, c^*)$
  return $\perp$
else if $(\mathsf{id}, \bar{v}) = (\mathsf{id}^*, v)$
  return $\mathsf{SKE.Dec}(\mathsf{K}, \bar{c}_s)$
else $\bar{y} \leftarrow \mathsf{WPRF.F}(\mathsf{fk}, (\mathsf{id}, \bar{v}, \mathsf{vk}^*))$
  $\bar{\mathsf{K}} \leftarrow \mathsf{SKE.Gen}(1^\lambda; \bar{y})$
  return $\mathsf{SKE.Dec}(\bar{\mathsf{K}}, \bar{c}_s)$

Fig. 8: Game 3

2. ABOS-challenger does the following:
   (a) $(\mathsf{sk}_0, \mathsf{vk}_0) \leftarrow \mathsf{ABOS.Setup}(1^\lambda)$
   (b) $(\mathsf{sk}_1, \mathsf{vk}_1) \leftarrow \mathsf{ABOS.PuncSetup}(1^\lambda, m^*)$
   (c) $\tilde{b} \leftarrow \{0,1\}$
   (d) return $\mathsf{vk}_{\tilde{b}}$ to $\mathcal{B}_3$
3. generate $(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{WPRF.Gen}(1^\lambda, R)$
4. pick $v \leftarrow \{0,1\}^{2\lambda}$
5. set $y \leftarrow \mathsf{WPRF.F}(\mathsf{fk}, (\mathsf{id}^*, v, \mathsf{vk}_{\tilde{b}}))$
6. compute $\mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y)$
7. set $\mathsf{pp} = (\mathsf{ek}, \mathsf{vk}_{\tilde{b}})$ and send it to $\mathcal{A}$
8. $\mathcal{A}$ can ask the following queries for polynomial number of times:
   (a) *key query for* id: $\mathcal{B}_3$ uses it's signing oracle $\mathsf{ABOS.Sig}(\mathsf{sk}_{\tilde{b}}, \cdot)$ to get a signature σ of id and return $\mathsf{sk}_{\mathsf{id}} = (\mathsf{id}, \sigma)$ if $\mathsf{id} \neq \mathsf{id}^*$, else return $\perp$
   (b) *ciphertext query for* (id, c): $\mathcal{B}_3$ uses the function $O_{\mathsf{D},\mathsf{vk}_{\tilde{b}},\mathsf{K}}(\cdot)$ defined in Fig. 8 for ciphertext query of $\mathcal{A}$
9. $\mathcal{A}$ submits the challenge messages (m_0, m_1)
10. pick $b \leftarrow \{0,1\}$ and computes $c_s^* \leftarrow \mathsf{SKE.Enc}(\mathsf{K}, m_b)$
11. set $c^* = (c_s^*, v)$ and send it to $\mathcal{A}$
12. $\mathcal{A}$ may repeat the step 8 and returns a guess b' for b
13. return 1 if $b = b'$ and $|m_0| = |m_1|$

It is easy to see that if $\tilde{b} = 0$ then $\mathcal{B}_3$ simulates the KeyGen oracle $O_{\mathsf{sk}}(\cdot)$ of Game 2 and if $\tilde{b} = 1$ then $\mathcal{B}_3$ simulates the KeyGen oracle $O_{\mathsf{sk}^*}(\cdot)$ of Game 3. Next, we show that $O_{\mathsf{D},\mathsf{vk}_0,\mathsf{K}}(\cdot)$ works like the oracle $O_{\mathsf{D}}(\cdot)$ as in Game 2. For any arbitrary query $(\mathsf{id}, c = (\bar{c}_s, \bar{v}))$, let us consider the following cases

Case 1 $(\mathsf{id}, c) = (\mathsf{id}^*, c^*)$: Both the oracles return $\perp$ as it is not a valid query.

Case 2 $(\mathsf{id}, \bar{v}) = (\mathsf{id}^*, v) \wedge (\bar{c}_s \neq c_s^*)$: Let, $z_0 = (\mathsf{id}^*, v, \mathsf{vk}_0)$. The oracle $O_{\mathsf{D}}(\cdot)$ generates a signature $\sigma \leftarrow \mathsf{ABOS.Sign}(\mathsf{sk}_0, \mathsf{id}^*)$ (where $(\mathsf{sk}_0, \mathsf{vk}_0) \leftarrow \mathsf{ABOS.Setup}(1^\lambda)$ as in Game 2, Fig. 7) and uses $y \leftarrow \mathsf{WPRF.Eval}(\mathsf{ek}, z_0, \sigma)$ to generate the decryption key. On the other hand, $O_{\mathsf{D},\mathsf{vk}_0,\mathsf{K}}(\cdot)$ uses $y^* \leftarrow \mathsf{WPRF.F}(\mathsf{fk}, z_0)$ to generate the decryption key. By the correctness of Eval, $y^* = y$ as $R(z_0, \sigma) = 1$.

Case 3 $(\mathsf{id}, \bar{v}) \neq (\mathsf{id}^*, v)$: Let $z = (\mathsf{id}, \bar{v}, \mathsf{vk}_0)$. The oracle $O_{\mathsf{D}}(\cdot)$ generates a signature $\sigma \leftarrow \mathsf{ABOS.Sig}(\mathsf{sk}_0, \mathsf{id})$ (as in Game 2) and uses $y \leftarrow \mathsf{WPRF.Eval}(\mathsf{ek}, z, \sigma)$ to generate the decryption key. $O_{\mathsf{D},\mathsf{vk}_0,\mathsf{K}}(\cdot)$ uses $y \leftarrow \mathsf{WPRF.F}(\mathsf{fk}, z)$ to generate the decryption key. By the similar argument as in case 2, we conclude that both the oracles compute the same decryption key.

Thus, $\mathcal{B}_3$ perfectly simulates Game 2 when $\tilde{b} = 0$. On the other hand, when the ABOS challenger picks $\tilde{b} = 1$, it perfectly simulates Game 3. Therefore, the advantage of $\mathcal{A}$ in distinguishing between the games 2 and 3 is the same as wining advantage of $\mathcal{B}_3$ in $\mathsf{VK\text{-}IND}$ security experiment and we write it as $|\Pr[\mathsf{G}_2] - \Pr[\mathsf{G}_3]| = \mathsf{Adv}_{\mathcal{B}_3}^{\mathsf{ABOS}}(\lambda)$ which is negligible in λ by our assumption.

Lemma 2 *Assuming* WPRF *is a selectively secure witness pseudorandom function, we have* $|\Pr[\mathsf{G}_3] - \Pr[\mathsf{G}_4]| = \mathsf{negl}(\lambda)$.

Proof. We show that if $\mathcal{A}$ can distinguish between the games 3 and 4, then there exists an adversary $\mathcal{B}_4$ which will break the selective security of WPRF (Def. 7). The challenge statement for $\mathcal{B}_4$ is $z^* = (\mathsf{id}^*, v, \mathsf{vk}^*)$ where $v \leftarrow \{0,1\}^{2\lambda}$ and $(\mathsf{sk}^*, \mathsf{vk}^*) \leftarrow \mathsf{ABOS.PuncSetup}(1^\lambda, \mathsf{id}^*)$. Note that, $v \leftarrow \{0,1\}^{2\lambda}$ implies that there exits $u \in \{0,1\}^\lambda$ satisfying $\mathsf{PRG}(\mathsf{id}^* \oplus u) = v$ holds with a negligible probability of (at most) $2^{-\lambda}$. Furthermore, by the correctness of $\mathsf{PuncSetup}$ (Def. 4), we have $\mathsf{ABOS.Vrfy}(\mathsf{vk}^*, \mathsf{id}^*, \sigma) = 0$ for all $\sigma \in \Sigma$. Hence, $R(z^*, w) = 0$ holds with overwhelming probability for any $w \in \mathcal{W}$ and z^* is a valid challenge statement for $\mathcal{B}_4$. Below we describe how $\mathcal{B}_4$ simulates $\mathcal{A}$ using z^*.

$\underline{\mathcal{B}_4(1^\lambda, z^*)}$:
1. send z^* to its challenger
2. WPRF-challenger does the following:
   (a) generate $(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{WPRF.Gen}(1^\lambda, R)$
   (b) set $y_0 \leftarrow \mathsf{WPRF.F}(\mathsf{fk}, z^*)$ and $y_1 \leftarrow \mathcal{Y}$
   (c) pick $\tilde{b} \leftarrow \{0,1\}$
   (d) return $(\mathsf{ek}, y_{\tilde{b}})$ to $\mathcal{B}_4$
3. compute $\mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y_{\tilde{b}})$
4. set $\mathsf{pp} = (\mathsf{ek}, \mathsf{vk}^*)$ and send it to $\mathcal{A}$
5. $\mathcal{A}$ can query the following oracles for polynomial number of times:
   (a) *key query for* id: $\mathcal{B}_4$ uses the oracle $O_{\mathsf{sk}^*}(\cdot)$ as described in Fig. 9 to compute the secret-key for id
   (b) *ciphertext query for* (id, c): $\mathcal{B}_4$ uses the decryption oracle $O_{\mathsf{D}^*,\mathsf{vk}^*,\mathsf{K}}(\cdot)$ as defined in Fig. 9 to compute the message for the query (id, c)
6. $\mathcal{A}$ submits the challenge messages (m_0, m_1)
7. pick $b \leftarrow \{0,1\}$ and computes $c_s^* \leftarrow \mathsf{SKE.Enc}(\mathsf{K}, m_b)$
8. set $c^* = (c_s^*, v)$ and send it to $\mathcal{A}$
9. $\mathcal{A}$ may repeat the step 5 and returns a guess b' for b
10. return 1 if $b = b'$ and $|m_0| = |m_1|$

First, we note that the oracle $O_{\mathsf{sk}^*}(\cdot)$ remains the same as in Game 3. Next, we observe that if $\tilde{b} = 0$ then the decryption oracles $O_{\mathsf{D},\mathsf{vk}^*,\mathsf{K}}(\cdot)$ of Game 3 and

$\mathsf{id}^* \leftarrow \mathcal{A}(1^\lambda)$
$(\mathsf{sk}^*, \mathsf{vk}^*) \leftarrow \mathsf{ABOS.PuncSetup}(1^\lambda, \mathsf{id}^*)$
$(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{WPRF.Gen}(1^\lambda, R)$
set $\mathsf{pp} = (\mathsf{ek}, \mathsf{vk}^*)$ and $\mathsf{msk} = \mathsf{sk}^*$
$v \leftarrow \{0,1\}^{2\lambda}$
set $z^* = (\mathsf{id}^*, v, \mathsf{vk}^*)$
$\boxed{y \leftarrow \mathcal{Y}}$
$\mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y)$
$(m_0, m_1) \leftarrow \mathcal{A}^{O_{\mathsf{sk}^*}(\cdot), O_{\mathsf{D}^*,\mathsf{vk}^*,\mathsf{K}}(\cdot)}(\mathsf{pp})$
$b \leftarrow \{0,1\}$
$c_s^* \leftarrow \mathsf{SKE.Enc}(\mathsf{K}, m_b)$
return $c^* = (c_s^*, v)$
$b' \leftarrow \mathcal{A}^{O_{\mathsf{sk}^*}(\cdot), O_{\mathsf{D}^*,\mathsf{vk}^*,\mathsf{K}}(\cdot)}(c^*)$
return 1 if $(b' = b)$

$\underline{O_{\mathsf{sk}^*}(\cdot)}$:
input: $\mathsf{id} \in \mathcal{ID}$
compute $\sigma \leftarrow \mathsf{ABOS.Sig}(\mathsf{sk}^*, \mathsf{id})$
return $\mathsf{sk}_{\mathsf{id}} = (\mathsf{id}, \sigma)$ if $\mathsf{id} \neq \mathsf{id}^*$, else $\perp$

$\underline{O_{\mathsf{D}^*,\mathsf{vk}^*,\mathsf{K}}(\cdot)}$:
input: $(\mathsf{id} \in \mathcal{ID}, c)$
parse $c = (\bar{c}_s, \bar{v})$
if $(\mathsf{id}, c) = (\mathsf{id}^*, c^*)$
  return $\perp$
else if $(\mathsf{id}, \bar{v}) = (\mathsf{id}^*, v)$
  return $\mathsf{SKE.Dec}(\mathsf{K}, \bar{c}_s)$
else $\bar{y} \leftarrow O_{\mathsf{fk}}((\mathsf{id}, \bar{v}, \mathsf{vk}^*))$
  $\bar{\mathsf{K}} \leftarrow \mathsf{SKE.Gen}(1^\lambda; \bar{y})$
  return $\mathsf{SKE.Dec}(\bar{\mathsf{K}}, \bar{c}_s)$

Here $O_{\mathsf{fk}}(z) = \mathsf{WPRF.F}(\mathsf{fk}, z)$ if $z \neq z^*$, else $\perp$

Fig. 9: Game 4

$O_{\mathsf{D}^*,\mathsf{vk}^*,\mathsf{K}}(\cdot)$ of Game 4 are functionally equivalent. More precisely, for any arbitrary query $(\mathsf{id}, c = (\bar{c}_s, \bar{v}))$ we consider the following cases

Case 1 $(\mathsf{id}, c) = (\mathsf{id}^*, c^*)$: Both the oracles return $\perp$ as it is not a valid query.

Case 2 $(\mathsf{id}, \bar{v}) = (\mathsf{id}^*, v) \wedge (\bar{c}_s \neq c_s^*)$: Both the oracles $O_{\mathsf{D},\mathsf{vk}^*,\mathsf{K}}(\cdot)$ and $O_{\mathsf{D}^*,\mathsf{vk}^*,\mathsf{K}}(\cdot)$ utilize $y_0 \leftarrow \mathsf{WPRF.F}(\mathsf{fk}, z^*)$ to generate the decryption key.

Case 3 $(\mathsf{id}, \bar{v}) \neq (\mathsf{id}^*, v)$: Let $z = (\mathsf{id}, \bar{v}, \mathsf{vk}^*) \neq z^*$. Then, $O_{\mathsf{D},\mathsf{vk}^*,\mathsf{K}}(\cdot)$ computes $y \leftarrow \mathsf{WPRF.F}(\mathsf{fk}, z)$ to generate the decryption key. On the other hand, $O_{\mathsf{D}^*,\mathsf{vk}^*,\mathsf{K}}(\cdot)$ uses $y \leftarrow O_{\mathsf{fk}}(z)$ to generate the decryption key. Note that $O_{\mathsf{fk}}(z) = \mathsf{WPRF.F}(\mathsf{fk}, z)$ as $z \neq z^*$. Hence, both oracles compute the same decryption key.

Therefore, if the WPRF challenger picks the bit $\tilde{b} = 0$, then $y_{\tilde{b}} = \mathsf{WPRF.F}(\mathsf{fk}, (\mathsf{id}^*, v, \mathsf{vk}^*))$ and hence $\mathcal{B}_4$ simulates Game 3. If $\tilde{b} = 1$ then y is chosen uniformly at random from $\mathcal{Y}$ and hence $\mathcal{B}_4$ simulates Game 4. This implies that the advantage of $\mathcal{A}$ in distinguishing between the games 3 and 4 is the same as the advantage of $\mathcal{B}_4$ in the WPRF security experiment. Therefore, $|\Pr[\mathsf{G}_3] - \Pr[\mathsf{G}_4]| = \mathsf{Adv}_{\mathcal{B}_4}^{\mathsf{WPRF},R}(\lambda)$ which is negligible in λ by our assumption.

3.1 From IBE to CCA1 secure MIFHE

In this section, we describe our transformation from the above IBE to MIFHE. At first, we recall the definition of MFHE given by Mukherjee and Wichs [27] where they built a (pure) MFHE based on LWE along with circular security.

Definition 10 [27] A multi-key (pure) fully homomorphic encryption (MFHE) scheme is a tuple of PPT algorithms (Setup, KeyGen, Enc, Expand, Eval, Dec) defined as follows:

- $\mathsf{params} \leftarrow \mathsf{Setup}(1^\lambda)$: on input a security parameter λ, produces a system parameter params (which implicitly available to all other algorithms).

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{params})$: on input a system parameter params, outputs a secret-key sk and a public-key pk.
- $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$: returns c, a *fresh* ciphertext for a message $m \in \{0,1\}$.
- $\widehat{c} \leftarrow \mathsf{Expand}((\mathsf{pk}_1, \ldots, \mathsf{pk}_N), i, c)$: a deterministic algorithm that on input a sequence of N public-keys $(\mathsf{pk}_1, \ldots, \mathsf{pk}_N)$ and a fresh ciphertext c encrypted under the i^{th} key pk_i, returns an *expanded* ciphertext $\widehat{c}$.
- $\widehat{c} \leftarrow \mathsf{Eval}(\mathsf{params}, C, (\widehat{c}_1, \ldots, \widehat{c}_\ell))$: a deterministic algorithm that on input a polynomial-size boolean circuit C and a sequence of ℓ expanded ciphertexts $(\widehat{c}_1, \ldots, \widehat{c}_\ell)$, outputs an *evaluated* ciphertext $\widehat{c}$.
- $\mathsf{Dec}(\mathsf{params}, (\mathsf{sk}_1, \ldots, \mathsf{sk}_N), c) \in \{0,1\} \cup \{\perp\}$: a deterministic algorithm that on input N secret-keys $\mathsf{sk}_1, \ldots, \mathsf{sk}_N$ and a ciphertext c, returns either a message $m \in \{0,1\}$ or $\perp$ if it fails.

The MFHE is said to be correct and compact if the following holds:

For $\mathsf{params} \leftarrow \mathsf{Setup}(1^\lambda)$, $\{(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}(\mathsf{params})\}_{i \in [N]}$ and any ℓ-tuple message $(m_1, \ldots, m_\ell) \in \{0,1\}^\ell$, any sequence of indices $(I_1, \ldots, I_\ell) \in [N]^\ell$, $\{c_i \leftarrow \mathsf{Enc}(\mathsf{pk}_{I_i}, m_i)\}_{i \in [\ell]}$, $\{\widehat{c}_i \leftarrow \mathsf{Expand}((\mathsf{pk}_1, \ldots, \mathsf{pk}_N), I_i, c_i)\}_{i \in [\ell]}$ and a polynomial-size boolean circuit C, we have

- *correctness of* Expand: $\mathsf{Dec}(\mathsf{params}, (\mathsf{sk}_1, \ldots, \mathsf{sk}_N), \widehat{c}_i) = m_i$ for all $i \in [\ell]$.
- *correctness of* Eval: $\mathsf{Dec}(\mathsf{params}, (\mathsf{sk}_1, \ldots, \mathsf{sk}_N), \widehat{c}) = C(m_1, \ldots, m_\ell)$ where $\widehat{c} \leftarrow \mathsf{Eval}(\mathsf{params}, C, (\widehat{c}_1, \ldots, \widehat{c}_\ell))$.
- *compactness*: The size of an evaluated ciphertext $|\widehat{c}|$ is bounded by a fixed polynomial $\mathsf{p}(\lambda, N)$ independent of the circuit C.

Definition 11 A MFHE scheme is said to be semantically secure if, for all PPT adversary $\mathcal{A}$ and $\mathsf{params} \leftarrow \mathsf{Setup}(1^\lambda)$, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{params})$, any pair of messages $(m_0, m_1) \in \{0,1\}^2$, there exists a negligible function negl such that

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{MFHE}}(\lambda) = |\Pr[\mathcal{A}(\mathsf{params}, \mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, m_0)) = 1] - \Pr[\mathcal{A}(\mathsf{params}, \mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, m_1)) = 1] \,| < \mathsf{negl}(\lambda)$$

Definition 12 [8] A multi-identity (pure) fully homomorphic encryption (MIFHE) scheme is a tuple of PPT algorithms (Setup, KeyGen, Enc, Eval, Dec) where Setup, KeyGen and Enc are the same as in a normal IBE scheme (Def. 8) and the remaining two algorithms work as follows:

- $\widehat{c} \leftarrow \mathsf{Eval}(\mathsf{pp}, C, (c_1, \ldots, c_\ell))$: a deterministic algorithm that on input a public parameter pp, a polynomial-size boolean circuit C and ciphertexts $c_1, \ldots, c_\ell$ (each of which encrypts a bit using Enc), outputs an evaluated ciphertext $\widehat{c}$.
- $\mathsf{Dec}(\mathsf{pp}, (\mathsf{sk}_{\mathsf{id}_1}, \ldots, \mathsf{sk}_{\mathsf{id}_\ell}), c) \in \{0,1\} \cup \{\perp\}$: a deterministic algorithm that on input a public parameter pp, ℓ secret-keys $\mathsf{sk}_{\mathsf{id}_1}, \ldots, \mathsf{sk}_{\mathsf{id}_\ell}$ corresponding to the identities $\mathsf{id}_1, \ldots, \mathsf{id}_\ell$ and a ciphertext c encrypted under the identities $\mathsf{id}_1, \ldots, \mathsf{id}_\ell$, outputs either a message $m \in \{0,1\}$ or $\perp$ if it fails.

The MIFHE is said to be correct and compact if the following hold:

- *correctness*: For $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$, $\{\mathsf{sk}_{\mathsf{id}_i} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{id}_i)\}_{i \in [\ell]}$ and any ℓ-tuple message $(m_1, \ldots, m_\ell) \in \{0,1\}^\ell$ such that $\{c_i \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{id}_i, m_i)\}_{i \in [\ell]}$ and a polynomial-size boolean circuit C, we have

$\underline{\mathsf{Setup}(1^\lambda)}$:

$(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{ABOS.Setup}(1^\lambda)$
$(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{WPRF.Gen}(1^\lambda, R)$
$\mathsf{params} \leftarrow \mathsf{MFHE.Setup}(1^\lambda)$
set $\mathsf{pp} = (\mathsf{ek}, \mathsf{vk}, \mathsf{params})$, $\mathsf{msk} = \mathsf{sk}$
return $(\mathsf{pp}, \mathsf{msk})$

$\underline{\mathsf{KeyGen}(\mathsf{msk}, \mathsf{id})}$:

parse $\mathsf{msk} = \mathsf{sk}$
$\sigma \leftarrow \mathsf{ABOS.Sig}(\mathsf{sk}, \mathsf{id})$
set $\mathsf{sk}_{\mathsf{id}} = (\sigma, \mathsf{id})$
return $\mathsf{sk}_{\mathsf{id}}$

$\underline{\mathsf{Eval}(\mathsf{pp}, C, (\mathsf{ct}_1, \ldots, \mathsf{ct}_\ell))}$:

parse $\mathsf{pp} = (\mathsf{ek}, \mathsf{vk}, \mathsf{params})$
parse $\mathsf{ct}_i = (c_{v_i}, v_i, \mathsf{pk}_{v_i})$, $\forall i \in [\ell]$
for $i = 1$ to ℓ
  $\widehat{c}_i \leftarrow \mathsf{MFHE.Expand}((\mathsf{pk}_{v_1}, \ldots, \mathsf{pk}_{v_\ell}), i, c_{v_i})$
$\widehat{c} \leftarrow \mathsf{MFHE.Eval}(\mathsf{params}, C, (\widehat{c}_1, \ldots, \widehat{c}_\ell))$
return $\widehat{\mathsf{ct}} = (\widehat{c}, \{v_i, \mathsf{pk}_{v_i}\}_{i \in [\ell]})$

$\underline{\mathsf{Enc}(\mathsf{pp}, \mathsf{id}, m)}$:

parse $\mathsf{pp} = (\mathsf{ek}, \mathsf{vk}, \mathsf{params})$
$u \leftarrow \{0,1\}^\lambda$, $v \leftarrow \mathsf{PRG}(\mathsf{id} \oplus u)$
$y_v \leftarrow \mathsf{WPRF.Eval}(\mathsf{ek}, (\mathsf{id}, v, \mathsf{vk}), u)$
$(\mathsf{pk}_v, \mathsf{sk}_v) \leftarrow \mathsf{MFHE.KeyGen}(\mathsf{params}; y_v)$
$c_v \leftarrow \mathsf{MFHE.Enc}(\mathsf{pk}_v, m)$
return $\mathsf{ct} = (c_v, v, \mathsf{pk}_v)$

$\underline{\mathsf{Dec}(\mathsf{pp}, (\mathsf{sk}_{\mathsf{id}_1}, \ldots, \mathsf{sk}_{\mathsf{id}_\ell}), \widehat{\mathsf{ct}})}$:

parse $\mathsf{pp} = (\mathsf{ek}, \mathsf{vk}, \mathsf{params})$
parse $\mathsf{sk}_{\mathsf{id}_i} = (\sigma_i, \mathsf{id}_i)$, $\forall i \in [\ell]$
parse $\widehat{\mathsf{ct}} = (\widehat{c}, \{v_i, \mathsf{pk}_{v_i}\}_{i \in [\ell]})$
for $i = 1$ to ℓ
  $y_i \leftarrow \mathsf{WPRF.Eval}(\mathsf{ek}, (\mathsf{id}_i, v_i, \mathsf{vk}), \sigma_i)$
  $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{MFHE.KeyGen}(\mathsf{params}; y_i)$
  if $\mathsf{pk}_i \neq \mathsf{pk}_{v_i}$
    return $\perp$
return $\mathsf{MFHE.Dec}(\mathsf{params}, (\mathsf{sk}_1, \ldots, \mathsf{sk}_\ell), \widehat{c})$

Fig. 10: Construction of multi-identity pure FHE

$$\Pr[\mathsf{Dec}(\mathsf{pp}, (\mathsf{sk}_{\mathsf{id}_1}, \ldots, \mathsf{sk}_{\mathsf{id}_\ell}), \mathsf{Eval}(\mathsf{pp}, C, (c_1, \ldots, c_\ell))) = C(m_1, \ldots, m_\ell)] = 1$$

– *compactness*: The size of an evaluated ciphertext $|\widehat{c}|$ is bounded by a fixed polynomial $\mathsf{p}(\lambda, N)$ independent of the circuit C.

We consider $\mathsf{CCA1}$ security for MIFHE where the adversary has an access to the decryption oracle before it receives the challenge ciphertext. We skip the formal description of the security as it is almost similar to Def. 9.

Construction. We construct a multi-identity pure FHE scheme $\mathsf{MIFHE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ for an identity space $\mathcal{ID} = \{0,1\}^\lambda$, a message space $\{0,1\}$ and a class of polynomial sized circuits $\{\mathcal{C}_\lambda\}$. We consider the same set of primitives that are employed in the basic IBE of Sec. 3 except SKE is replaced by a pure MFHE scheme. Our MIFHE is described in Fig. 10. The correctness is followed by a similar argument as in our IBE scheme and using the correctness of MFHE scheme. We state the security in the following theorem.

Theorem 2 *The* $\mathsf{MIFHE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ *described in Figure 10 is a selective-identity* $\mathsf{CCA1}$ *secure multi-identity pure fully homomorphic encryption if* PRG *is a secure pseudorandom generator,* WPRF *is a selectively secure puncturable witness pseudorandom function,* ABOS *is a* $\mathsf{VK\text{-}IND}$ *secure all-but-one signature scheme and* MFHE *is a semantically secure multi-key pure fully homomorphic encryption.*

Proof. The proof is similar to the Th. 1 with few changes. Firstly, we replace SKE with MFHE. Secondly, observe that the semantic security of MFHE is sufficient as we consider $\mathsf{CCA1}$ security for which $\mathcal{A}$ is not allowed to query the decryption oracle after the challenge query. More specifically, the secret-key sk_v, associated with the public-key pk_v which encrypts the challenge message, is no longer needed for any decryption oracle used in the proof. This is due to the fact that after Game 2 the component v of the challenge ciphertext (c_v, v, pk_v) is chosen uniformly from $\{0,1\}^{2\lambda}$ and hence for all the decryption queries $\{(\mathsf{id}, (\bar{c}_v, \bar{v}, \bar{\mathsf{pk}}_v))\}$ of $\mathcal{A}$ we have $v \neq \bar{v}$ with overwhelming probability. Thus, we omit the lines 5 and 6 from both the oracles $O_{\mathsf{D},\mathsf{vk}^*,\mathsf{K}}$ and $O_{\mathsf{D}^*,\mathsf{vk}^*,\mathsf{K}}$, and rename them by $O_{\mathsf{D},\mathsf{vk}^*}$ and $O_{\mathsf{D}^*,\mathsf{vk}^*}$ respectively. Finally, at the end of Game 4 we generate the key pair $(\mathsf{pk}_v, \mathsf{sk}_v) \leftarrow \mathsf{MFHE.KeyGen}(\mathsf{params}; y_v)$ using a fresh randomness y_v which is independent of the challenge identity id^*. Therefore, the semantic security of MFHE guarantees that $(\mathsf{MFHE.Enc}(\mathsf{pk}_v, 0),\ v,\ \mathsf{pk}_v)$ is indistinguishable from $(\mathsf{MFHE.Enc}(\mathsf{pk}_v, 1),\ v,\ \mathsf{pk}_v)$ which completes the proof.

4 CCA1 Secure MAFHE from WPRF and MFHE

In this section, we present a construction of a CCA1 secure multi-attribute pure FHE (MAFHE) using WPRF and MFHE. The heart of our MAFHE is a CCA secure (key-policy) ABE. We start with the definition of ABE.

Definition 13 [32] An attribute-based encryption (ABE) scheme for a class of functions $\{\mathcal{F}_\lambda\}$ is a tuple of PPT algorithms (Setup, KeyGen, Enc, Dec) defined as follows:

- $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$: on input a security parameter λ, produces a public parameter pp and a master secret-key msk.
- $\mathsf{sk}_f \leftarrow \mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, f)$: returns a secret-key sk_f corresponding to the function $f \in \mathcal{F}_\lambda$.
- $c \leftarrow \mathsf{Enc}(\mathsf{pp}, x, m)$: returns c, an encryption of a message $m \in \mathcal{M}$ under an attribute $x \in \mathcal{X}$.
- $\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_f, c) \in \mathcal{M} \cup \{\perp\}$: a deterministic algorithm that decrypts a ciphertext c using sk_f and outputs either a message $m \in \mathcal{M}$ or $\perp$ if it fails.

The ABE is said to be correct if the following holds:

– *correctness*: For all $f \in \mathcal{F}_\lambda$, $x \in \mathcal{X}$, $m \in \mathcal{M}$, $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and $\mathsf{sk}_f \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{id})$, we require that

$$\Pr[\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_f, \mathsf{Enc}(\mathsf{pp}, x, m) = m : f(x) = 1] = 1$$

We consider selective-attribute CCA security for ABE and define the security experiment $\mathsf{Expt}^{\mathsf{ABE}}_{\mathcal{A},\mathsf{CCA}}(1^\lambda)$ in Fig. 11.

Definition 14 An attribute-based encryption ABE is said to be selective-attribute CCA secure if, for all PPT adversary $\mathcal{A}$, there exists a negligible function negl such that

$$\mathsf{Adv}^{\mathsf{ABE}}_{\mathcal{A},\mathsf{CCA}}(\lambda) = |\Pr[\mathsf{Expt}^{\mathsf{ABE}}_{\mathcal{A},\mathsf{CCA}}(1^\lambda) = 1] - \tfrac{1}{2}| < \mathsf{negl}(\lambda)$$

Construction. We construct a selective-attribute CCA secure ABE based on the ABE of [15] which was built using witness encryption. The following ingredients are utilized:

1. $x^* \leftarrow \mathcal{A}(1^\lambda)$ 2. $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ 3. $(m_0, m_1) \leftarrow \mathcal{A}^{O_{\mathsf{sk}}(\cdot), O_{\mathsf{D}}(\cdot)}(\mathsf{pp})$ 4. $b \leftarrow \{0,1\}$ 5. $c^* \leftarrow \mathsf{Enc}(\mathsf{pp}, x^*, m_b)$ 6. $b' \leftarrow \mathcal{A}^{O_{\mathsf{sk}}(\cdot), O_{\mathsf{D}}(\cdot)}(c^*)$ 7. return 1 if $(b' = b) \wedge (\lvert m_0 \rvert = \lvert m_1 \rvert)$	$\underline{O_{\mathsf{sk}}(\cdot)}$: 1. input: $f \in \mathcal{F}_\lambda$ 2. compute $\mathsf{sk}_f \leftarrow \mathsf{KeyGen}(\mathsf{msk}, f)$ 3. return sk_f if $f(x^*) = 0$, else $\perp$ $\underline{O_{\mathsf{D}}(\cdot)}$: 1. input: $(f \in \mathcal{F}_\lambda,\ c)$ 2. compute $\mathsf{sk}_f \leftarrow \mathsf{KeyGen}(\mathsf{msk}, f)$ 3. return $\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_f, c)$ unless $(f, c) = (f, c^*) \wedge f(x^*) = 1$, else $\perp$

Fig. 11: $\mathsf{Expt}^{\mathsf{ABE}}_{\mathcal{A},\mathsf{CCA}}(1^\lambda)$

- A pseudorandom generator $\mathsf{PRG} : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$.
- A LP-CCA secure symmetric key encryption SKE = (Gen, Enc, Dec).
- A perfectly binding and computationally hiding commitment scheme $\mathsf{Com}(\cdot)$.
- A non-interactive zap = (Prv, Vrfy) for the NP language $L' = \{(\eta_1, \eta_2, f) : (\exists w_1 \text{ such that } \eta_1 = \mathsf{Com}(0; w_1)) \text{ or } (\exists (w_2, x) \text{ such that } \eta_2 = \mathsf{Com}(0^n; w_2) \wedge f(x) = 0)\}$ (see the Def. 16 of App. A).
- A WPRF = (Gen, F, Eval) for the NP language $L = \{(x, v) : (\exists\ u \in \{0,1\}^\lambda \text{ such that } \mathsf{PRG}(x \oplus u) = v) \text{ or } (\exists (\eta_1, \eta_2, f, \pi) \text{ such that } \mathsf{Vrfy}((\eta_1, \eta_2, f), \pi) = 1 \wedge f(x) = 1)\}$ with a relation $R : \mathcal{X} \times \mathcal{W} \to \{0,1\}$.

We describe our construction in Fig. 12. For *correctness*, we notice that whenever $f(x) = 1$ holds (η_1, η_2, f, π) becomes a valid witness of the statement (x, v) corresponding to the relation R of WPRF where $\pi \leftarrow \mathsf{zap.Prv}((\eta_1, \eta_2, f), r)$. In other words, $\mathsf{zap.Vrfy}((\eta_1, \eta_2, f), \pi) = 1$ and we have

$$\begin{aligned} \mathsf{WPRF.F}(\mathsf{fk}, (x, v)) &= \mathsf{WPRF.Eval}(\mathsf{ek}, (x, v), (\eta_1, \eta_2, f, \pi)) && \text{[Decyption]} \\ &= \mathsf{WPRF.Eval}(\mathsf{ek}, (x, v), u) && \text{[Encryption]} \end{aligned}$$

Therefore, the same randomness is used to obtain the SKE key during encryption and decryption if $f(x) = 1$ and the original message can be recovered from $\widehat{c}$. The key efficiency factor is that the size of ciphertext (excluding the size of the attribute) is $|c| = |c_x| + |v| = |m| + 2\lambda$ which is *optimal* for any ABE scheme. Note that, plaintext and ciphertext sizes are the same for the SKE encryption.

Theorem 3 *The* ABE = (Setup, KeyGen, Enc, Dec) *described in Figure 12 is a selective-attribute* CCA *secure attribute-based encryption if* PRG *is a secure pseudorandom generator,* Com *is a perfectly binding and computationally hiding commitment scheme,* zap *is a non-interactive zap,* WPRF *is a selectively secure puncturable witness pseudorandom function and* SKE *is a* LP-CCA *secure symmetric key encryption.* (The proof is shifted to App. A.1)

4.1 From ABE to CCA1 Secure MAFHE

This section is devoted to present a CCA1 secure multi-attribute pure FHE (MAFHE) using the technique involved in our ABE and a multi-key pure FHE. At first, we state a formal definition of MAFHE.

$\underline{\mathsf{Setup}(1^\lambda)}$:

1. $(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{WPRF.Gen}(1^\lambda, R)$
2. $\eta_1 = \mathsf{Com}(0; r), \eta_2 = \mathsf{Com}(0^\lambda; s)$
3. set $\mathsf{pp} = (\mathsf{ek}, \eta_1, \eta_2)$, $\mathsf{msk} = r$
4. return $(\mathsf{pp}, \mathsf{msk})$

$\underline{\mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, \mathsf{id})}$:

1. parse $\mathsf{pp} = (\mathsf{ek}, \eta_1, \eta_2)$, $\mathsf{msk} = r$
2. $\pi_f \leftarrow \mathsf{zap.Prv}((\eta_1, \eta_2, f), r)$
3. set $\mathsf{sk}_f = (f, \pi_f)$
4. return sk_f

$\underline{\mathsf{Enc}(\mathsf{pp}, x, m)}$:

1. parse $\mathsf{pp} = (\mathsf{ek}, \eta_1, \eta_2)$
2. $u \leftarrow \{0,1\}^\lambda$, $v \leftarrow \mathsf{PRG}(x \oplus u)$
3. $y \leftarrow \mathsf{WPRF.Eval}(\mathsf{ek}, (x, v), u)$
4. $\mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y)$
5. $c_x \leftarrow \mathsf{SKE.Enc}(\mathsf{K}, m)$
6. return $c = (x, c_x, v)$

$\underline{\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_f, c)}$:

1. parse $\mathsf{pp} = (\mathsf{ek}, \eta_1, \eta_2)$
2. parse $\mathsf{sk}_f = (f, \pi)$, $c = (x, \widehat{c}, v)$
3. $y \leftarrow \mathsf{WPRF.Eval}(\mathsf{ek}, (x, v), (\eta_1, \eta_2, f, \pi))$
4. $\mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y)$
5. return $\mathsf{SKE.Dec}(\mathsf{K}, \widehat{c})$

Fig. 12: Construction of ABE with optimal ciphertexts

Definition 15 A multi-attribute (pure) fully homomorphic encryption (MAFHE) scheme for a function class $\{\mathcal{F}_\lambda\}$ and an attribute space $\mathcal{X}$ is a tuple of PPT algorithms (Setup, KeyGen, Enc, Eval, Dec) where Setup, KeyGen and Enc are the same as in a normal ABE scheme (Def. 13). The remaining two algorithms work as follows:

- $\widehat{c} \leftarrow \mathsf{Eval}(\mathsf{pp}, C, (c_1, \ldots, c_\ell))$: a deterministic algorithm that on input a public parameter pp, a boolean circuit C of polynomial size and ciphertexts $c_1, \ldots, c_\ell$ (each of which encrypts a bit using Enc), outputs an evaluated ciphertext $\widehat{c}$.
- $\mathsf{Dec}(\mathsf{pp}, (\mathsf{sk}_{f_1}, \ldots, \mathsf{sk}_{f_\ell}), c) \in \{0,1\} \cup \{\perp\}$: a deterministic algorithm that on input a public parameter pp, a sequence of secret-keys $(\mathsf{sk}_{f_1}, \ldots, \mathsf{sk}_{f_\ell})$ corresponding to the functions $f_1, \ldots, f_\ell \in \mathcal{F}_\lambda$ and a ciphertext c encrypted under the attributes $x_1, \ldots, x_\ell \in \mathcal{X}$, outputs either a message $m \in \{0,1\}$ or $\perp$ if it fails.

The MAFHE is said to be correct and compact if the following hold:

– *correctness*: For $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$, $\{\mathsf{sk}_{f_i} \leftarrow \mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, f_i)\}_{i \in [\ell]}$ and any ℓ-tuple messages $(m_1, \ldots, m_\ell) \in \{0,1\}^\ell$ such that $\{c_i \leftarrow \mathsf{Enc}(\mathsf{pp}, x_i, m_i)\}_{i \in [\ell]}$ satisfying $f_i(x_i) = 1\ \forall i \in [\ell]$ and a boolean circuit C of polynomial size, we have

$$\Pr[\mathsf{Dec}(\mathsf{pp}, (\mathsf{sk}_{f_1}, \ldots, \mathsf{sk}_{f_\ell}), \mathsf{Eval}(\mathsf{pp}, C, (c_1, \ldots, c_\ell))) = C(m_1, \ldots, m_\ell)] = 1$$

– *compactness*: There exists a fixed polynomial $\mathsf{p}(\cdot)$ such that the size of an evaluated ciphertext is bounded by $\mathsf{p}(\lambda)$. This means $|\widehat{c}|$ does not depend on the circuit C.

We consider CCA1 security for MAFHE where the adversary is given access to the decryption oracle until it receives the challenge ciphertext. We skip the formal description of the security as it is almost similar to Def. 14 where the decryption oracle is not provided after generating the challenge ciphertext.

$\underline{\mathsf{Setup}(1^\lambda)}$:
1. $(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{WPRF.Gen}(1^\lambda, R)$
2. $\eta_1 = \mathsf{Com}(0; r), \eta_2 = \mathsf{Com}(0^\lambda; s)$
3. $\mathsf{params} \leftarrow \mathsf{MFHE.Setup}(1^\lambda)$
4. set $\mathsf{pp} = (\mathsf{ek}, \eta_1, \eta_2, \mathsf{params})$, $\mathsf{msk} = r$
5. return $(\mathsf{pp}, \mathsf{msk})$

$\underline{\mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, f)}$:
1. parse $\mathsf{pp} = (\mathsf{ek}, \eta_1, \eta_2, \mathsf{params})$, $\mathsf{msk} = r$
2. $\pi_f \leftarrow \mathsf{zap.Prv}((\eta_1, \eta_2, f), r)$
3. set $\mathsf{sk}_f = (f, \pi_f)$
4. return sk_f

$\underline{\mathsf{Eval}(\mathsf{pp}, C, (\mathsf{ct}_1, \ldots, \mathsf{ct}_\ell))}$:
1. parse $\mathsf{pp} = (\mathsf{ek}, \eta_1, \eta_2, \mathsf{params})$
2. parse $\mathsf{ct}_i = (c_{v_i}, x_i, v_i, \mathsf{pk}_{v_i})$, $\forall i \in [\ell]$
3. for $i = 1$ to ℓ
4. $\quad \widehat{c}_i \leftarrow \mathsf{MFHE.Expand}((\mathsf{pk}_{v_1}, \ldots, \mathsf{pk}_{v_\ell}), i, c_{v_i})$
5. $\widehat{c} \leftarrow \mathsf{MFHE.Eval}(\mathsf{params}, C, (\widehat{c}_1, \ldots, \widehat{c}_\ell))$
6. return $\widehat{\mathsf{ct}} = (\widehat{c}, \{x_i, v_i, \mathsf{pk}_{v_i}\}_{i\in[\ell]})$

$\underline{\mathsf{Enc}(\mathsf{pp}, x, m)}$:
1. parse $\mathsf{pp} = (\mathsf{ek}, \eta_1, \eta_2, \mathsf{params})$
2. $u \leftarrow \{0,1\}^\lambda$, $v \leftarrow \mathsf{PRG}(x \oplus u)$
3. $y_v \leftarrow \mathsf{WPRF.Eval}(\mathsf{ek}, (x, v), u)$
4. $(\mathsf{pk}_v, \mathsf{sk}_v) \leftarrow \mathsf{MFHE.KeyGen}(\mathsf{params}; y_v)$
5. $c_v \leftarrow \mathsf{MFHE.Enc}(\mathsf{pk}_v, m)$
6. return $\mathsf{ct} = (c_v, x, v, \mathsf{pk}_v)$

$\underline{\mathsf{Dec}(\mathsf{pp}, (\mathsf{sk}_{f_1}, \ldots, \mathsf{sk}_{f_\ell}), \widehat{\mathsf{ct}})}$:
1. parse $\mathsf{pp} = (\mathsf{ek}, \eta_1, \eta_2, \mathsf{params})$
2. parse $\mathsf{sk}_{f_i} = (f_i, \pi_i)$, $\forall i \in [\ell]$
3. parse $\widehat{\mathsf{ct}} = (\widehat{c}, \{x_i, v_i, \mathsf{pk}_{v_i}\}_{i\in[\ell]})$
4. for $i = 1$ to ℓ
5. $\quad y_i \leftarrow \mathsf{WPRF.Eval}(\mathsf{ek}, (x_i, v_i), (\eta_1, \eta_2, f_i, \pi_i))$
6. $\quad (\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{MFHE.KeyGen}(\mathsf{params}; y_i)$
7. $\quad$ if $\mathsf{pk}_i \neq \mathsf{pk}_{v_i}$
8. $\quad\quad$ return $\perp$
9. return $\mathsf{MFHE.Dec}(\mathsf{params}, (\mathsf{sk}_1, \ldots, \mathsf{sk}_\ell), \widehat{c})$

Fig. 13: Construction of multi-attribute pure FHE

Construction. We are all set to describe a MAFHE scheme based on our ABE. The idea is similar to how we built the MIFHE from our IBE. Consequently, we need the same set of primitives as required in the ABE of Sec. 4 except the SKE is replaced by a semantically secure pure MFHE. The MAFHE for a function class $\{\mathcal{F}_\lambda\}$ and message space $\{0,1\}$ is described in Fig. 13. Note that, the setup algorithm does not take into account any predefined depth of supported circuits as we assume circular security of the underlying MFHE. The correctness can be similarly argued as in our ABE scheme along with the correctness of MFHE. The CCA1 security of our MAFHE is followed from the proof of Th. 3.

Theorem 4 *The* MAFHE = (Setup, KeyGen, Enc, Eval, Dec) *described in Figure 13 is a selective-attribute* CCA1 *secure multi-attribute pure fully homomorphic encryption if* PRG *is a secure pseudorandom generator,* Com *is a perfectly binding and computationally hiding commitment scheme,* zap *is a non-interactive zap,* WPRF *is a selectively secure puncturable witness pseudorandom function and* MFHE *is a semantically secure multi-key pure fully homomorphic encryption.* (The proof is discussed in App. A.2)

5 Conclusion

We propose two generic approaches to construct IBE and ABE from WPRF, both of which are CCA secure and achieve a ciphertext of size $|m| + 2\lambda$. Existing schemes do not satisfy such optimal ciphertext size along with CCA security. Additionally, with the help of a pure MFHE, we convert our IBE and ABE into CCA1 secure MIFHE and MAFHE schemes respectively. Existing MIFHE and MAFHE [11] are CPA secure and rely on (possibly stronger assumption of) $i\mathcal{O}$.

References

1. M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudorandom bits. *SIAM journal on Computing*, 13(4):850–864, 1984.
2. D. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. *SIAM Journal on Computing*, 36(5):1301–1328, 2007.
3. D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *Annual international cryptology conference*, pages 213–229. Springer, 2001.
4. E. Boyle, S. Goldwasser, and I. Ivan. Functional signatures and pseudorandom functions. In *International Workshop on Public Key Cryptography*, pages 501–519. Springer, 2014.
5. Z. Brakerski, D. Cash, R. Tsabary, and H. Wee. Targeted homomorphic attribute-based encryption. In *Theory of Cryptography Conference*, pages 330–360. Springer, 2016.
6. Z. Brakerski and V. Vaikuntanathan. Fully homomorphic encryption from ring-lwe and security for key dependent messages. In *Annual cryptology conference*, pages 505–524. Springer, 2011.
7. Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. *SIAM Journal on Computing*, 43(2):831–871, 2014.
8. R. Canetti, S. Raghuraman, S. Richelson, and V. Vaikuntanathan. Chosen-ciphertext secure fully homomorphic encryption. In *IACR International Workshop on Public Key Cryptography*, pages 213–240. Springer, 2017.
9. Y. Chen and Z. Zhang. Publicly evaluable pseudorandom functions and their applications. *Journal of Computer Security*, 24(2):289–320, 2016.
10. M. Clear and C. M. Goldrick. Attribute-based fully homomorphic encryption with a bounded number of inputs. *International Journal of Applied Cryptography*, 3(4):363–376, 2017.
11. M. Clear and C. McGoldrick. Bootstrappable identity-based fully homomorphic encryption. In *International Conference on Cryptology and Network Security*, pages 1–19. Springer, 2014.
12. M. Clear and C. McGoldrick. Multi-identity and multi-key leveled fhe from learning with errors. In *Annual Cryptology Conference*, pages 630–656. Springer, 2015.
13. R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 45–64. Springer, 2002.
14. J. Daemen and V. Rijmen. *The design of Rijndael: AES-the advanced encryption standard*. Springer Science & Business Media, 2013.
15. S. Garg, C. Gentry, A. Sahai, and B. Waters. Witness encryption and its applications. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 467–476. ACM, 2013.
16. C. Gentry. Practical identity-based encryption without random oracles. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 445–464. Springer, 2006.
17. C. Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 169–178, 2009.
18. C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Annual Cryptology Conference*, pages 75–92. Springer, 2013.

19. S. Goldwasser, Y. T. Kalai, R. A. Popa, V. Vaikuntanathan, and N. Zeldovich. How to run turing machines on encrypted data. In *Advances in Cryptology–CRYPTO 2013*, pages 536–553. Springer, 2013.
20. R. Goyal, S. Vusirikala, and B. Waters. Collusion resistant broadcast and trace from positional witness encryption. In *IACR International Workshop on Public Key Cryptography*, pages 3–33. Springer, 2019.
21. V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 89–98, 2006.
22. J. Groth, R. Ostrovsky, and A. Sahai. Perfect non-interactive zero knowledge for np. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 339–358. Springer, 2006.
23. S. Halevi and P. Rogaway. A tweakable enciphering mode. In *Annual International Cryptology Conference*, pages 482–499. Springer, 2003.
24. S. Halevi and P. Rogaway. A parallelizable enciphering mode. In *Cryptographers Track at the RSA Conference*, pages 292–304. Springer, 2004.
25. E. Kiltz. Direct chosen-ciphertext secure identity-based encryption in the standard model with short ciphertexts, 2006.
26. S. Micali, M. Rabin, and S. Vadhan. Verifiable random functions. In *40th annual symposium on foundations of computer science (cat. No. 99CB37039)*, pages 120–130. IEEE, 1999.
27. P. Mukherjee and D. Wichs. Two round multiparty computation via multi-key fhe. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 735–763. Springer, 2016.
28. T. Pal and R. Dutta. Offline witness encryption from witness prf and randomized encoding in crs model. In *Australasian Conference on Information Security and Privacy*, pages 78–96. Springer, 2019.
29. J. Rompel. One-way functions are necessary and sufficient for secure signatures. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 387–394, 1990.
30. A. Sahai and B. Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 475–484. ACM, 2014.
31. M. Van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 24–43. Springer, 2010.
32. S. Yamada, N. Attrapadung, G. Hanaoka, and N. Kunihiro. Generic constructions for chosen-ciphertext secure attribute based encryption. In *International Workshop on Public Key Cryptography*, pages 71–89. Springer, 2011.
33. M. Zhandry. How to avoid obfuscation using witness prfs. In *Theory of Cryptography Conference*, pages 421–448. Springer, 2016.

A CCA Security of Our ABE and MAFHE

In this section, we proof the selective-attribute CCA security of our ABE described in Sec. 4. The adversary $\mathcal{A}$ will submit the challenge attribute before setup. $\mathcal{A}$ has two oracles. A secret-key oracle O_{sk} that on input a function $f \in \mathcal{F}_\lambda$ outputs $\mathsf{sk}_f \leftarrow \mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, f)$. The other one is a decryption oracle O_{D} that on input (f, c) first computes $\mathsf{sk}_f \leftarrow \mathsf{KeyGen}(\mathsf{pp}, \mathsf{msk}, f)$ and outputs

$\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_f, c)$. Note that, $\mathcal{A}$ can not query the challenge ciphertext c^* with a function f such that $f(x^*) = 1$ and all secret-key queries $\{f_i\}$ must satisfy $f_i(x^*) = 0$. We begin with the definition of a non-interactive zap.

Definition 16 [22] A non-interactive zap (or simply zap) for an NP language L with a relation R is a tuple of PPT algorithms (Prv, Vrfy) where Prv is an efficient prover that takes as input a statement x, a witness w and outputs a proof π and Vrfy is a verification algorithm which takes as input a statement-proof pair (x, π) and outputs 1 if π is a valid proof showing $x \in L$, otherwise 0. The algorithms also satisfy the following properties:

- *perfect completeness*: For any PPT adversary $\mathcal{A}$, it holds that

$$\Pr[(x, w) \leftarrow \mathcal{A}(1^\lambda); \pi \leftarrow \mathsf{Prv}(1^\lambda, x, w) : \mathsf{Vrfy}(x, \pi) = 1 \text{ if } R(x, w) = 1] = 1$$

- *perfect soundness*: For all $x \notin L$ and for all PPT adversary $\mathcal{A}$ we have

$$\Pr[\pi \leftarrow \mathcal{A}(1^\lambda, x) : \mathsf{Vrfy}(x, \pi) = 1] = 0$$

- *witness-indistinguishability*: For all non-uniform PPT (interactive) adversary $\mathcal{A}$ the difference between the following two probabilities is negligible

$$\Pr\Big[(x, w_0, w_1) \leftarrow \mathcal{A}(1^\lambda); \pi \leftarrow \mathsf{Prv}(1^\lambda, x, w_0) : \mathcal{A}(\pi) = 1 \wedge R(x, w_b) = 1 \text{ for } b \in \{0, 1\}\Big]$$

$$\text{and } \Pr\Big[(x, w_0, w_1) \leftarrow \mathcal{A}(1^\lambda); \pi \leftarrow \mathsf{Prv}(1^\lambda, x, w_1) : \mathcal{A}(\pi) = 1 \wedge R(x, w_b) = 1 \text{ for } b \in \{0, 1\}\Big]$$

A.1 Proof of Th. 3

Proof. Let us consider the following hybrid games. In each game we assume that the size of challenge messages are equal.

Game 0: It is the standard experiment denoted as $\mathsf{Expt}^{\mathsf{ABE}}_{\mathcal{A},\mathsf{CCA}}(1^\lambda)$. Let x^* be the challenge attribute and $c^* = (x^*, c_{x^*}, v^*)$ be the challenge ciphertext.

Game 1: It is the same experiment as $\mathsf{Game\ 0}$ except that we now compute $y_{x^*} \leftarrow \mathsf{pWPRF.F}(\mathsf{fk}, (x^*, v^*))$ instead of using $\mathsf{pWPRF.Eval}$. By the correctness of Eval, the ciphertext distributions are the same in both games.

Game 2: It is same as $\mathsf{Game\ 1}$ except that we pick v^* uniformly at random from $\{0, 1\}^{2\lambda}$ instead of computing $v^* \leftarrow \mathsf{PRG}(x^* \oplus u)$. Since u is chosen uniformly at random from $\{0, 1\}^\lambda$, the distribution of $x^* \oplus u$ is uniform over $\{0, 1\}^\lambda$. The security of PRG (Def. 1) implies that the games 1 and 2 are indistinguishable.

Game 3: It is exactly same as $\mathsf{Game\ 1}$ except that we set $\eta_2 = \mathsf{Com}(x^*; s)$ instead of committing to 0^λ in the setup. The computationally hiding property of Com ensures that $\mathsf{Game\ 2}$ and $\mathsf{Game\ 3}$ are indistinguishable.

Game 4: It is identical to $\mathsf{Game\ 3}$ except we change the key generation oracle $O_{\mathsf{sk}}(\cdot)$. Instead of using r to prove the statement (η_1, η_2, f), we use (s, x^*) as the witness where s is the randomness used to generate η_2. If $f(x^*) = 0$ then $O_{\mathsf{sk}}(f)$ returns (f, π_f) where $\pi_f \leftarrow \mathsf{zap.Prv}((\eta_1, \eta_2, f), (x^*, s))$ (however, $O_D(\cdot)$ still uses r to generate secret-keys). Note that, an adversary is only allowed to query such a function f that satisfies $f(x^*) = 0$. Since the statement remains the same, witness-indistinguishability property of zap ensures that the games 3 and 4 are indistinguishable.

Game 5: It is same as Game 4 except that we change $O_{\mathsf{D}}(\cdot)$ as follows where $\mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^{\lambda}; y^*)$ is the SKE key used to encrypt m_b:

$O_{\mathsf{D,K}}(\cdot)$:

1. input: $(f \in \mathcal{F}_{\lambda},\ c)$
2. parse $c = (x, \widehat{c}, v)$
3. if $(\widehat{c} = c^* \wedge f(x) = 1) \vee (\widehat{c} \neq c^* \wedge f(x) = 0)$
4.   return $\perp$
5. else if $(x, v) = (x^*, v^*) \wedge f(x) = 1$
6.   return $\mathsf{SKE.Dec}(\mathsf{K}, \widehat{c})$
7. else if $f(x) = 1$
8. $y \leftarrow \mathsf{WPRF.F}(\mathsf{fk}, (x, v))$
9. $\bar{\mathsf{K}} \leftarrow \mathsf{SKE.Gen}(1^{\lambda}; y)$
10.   return $\mathsf{SKE.Dec}(\bar{\mathsf{K}}, \widehat{c})$

To avoid secret-key generation, we use the secret function key fk to generate the decryption key of SKE. One can observe that the oracles O_{D} and $O_{\mathsf{D,K}}$ are functionally equivalent. Hence, the two games are indistinguishable.

Game 6: It is same Game 5 except the fact that we change η_1 to be a commitment of 1, instead of committing to 0. By the computationally hiding property of Com, Game 5 and Game 6 are indistinguishable.

Game 7: It is same as Game 6 except we chose y^* uniformly from $\mathcal{Y}$ (range of $\mathsf{WPRF.F}(\mathsf{fk}, \cdot)$) instead of setting it as $y^* \leftarrow \mathsf{WPRF.F}(\mathsf{fk}, (x^*, v^*))$. Also, we slightly modify the decryption oracle from $O_{\mathsf{D,K}}$ to $O_{\mathsf{D^*,K}}$ which now uses a function $O_{\mathsf{fk}}(\cdot)$ that on input z outputs $\mathsf{WPRF.F}(\mathsf{fk}, z)$ if $z \neq (x^*, v^*)$, otherwise returns $\perp$. That is, the change is in the line 8 of $O_{\mathsf{D,K}}$. We compute $y \leftarrow O_{\mathsf{fk}}((x, v))$ in the line 8 of $O_{\mathsf{D^*,K}}$. Again, we observe that these two decryption oracles are functionally equivalent by the definition of $O_{\mathsf{fk}}(\cdot)$.

In this game, we claim that the statement (x^*, v^*) does not have any witness corresponding to the relation R. Since v^* is uniformly chosen from $\{0,1\}^{2\lambda}$, it is very unlikely to get u such that $\mathsf{PRG}(x^* \oplus u) = v^*$. Therefore, $R((x^*, v^*), (\eta_1, \eta_2, f, \pi, u)) = 1$ means there exists a valid proof $\pi \leftarrow \mathsf{zap.Prv}((\eta_1, \eta_2, f), w)$ and $f(x^*) = 1$. Thus we should have either $\eta_1 = \mathsf{Com}(0; w)$ or $w = (x, w')$ satisfying $\eta_2 = \mathsf{Com}(0^{\lambda}; w')$ and $f(x) = 0$. Note that, η_1 is a commitment of 1 and η_2 is a commitment of x^*. Thus, by the statistical binding property of Com, there cannot exist a valid witness for (η_1, η_2, f). In other words, $\mathsf{zap.Vrfy}((\eta_1, \eta_2, f), \pi) = 0$ for all possible π. This ensures that $(x^*, v^*) \notin L$. By the similar argument as in Lemma 2 (of Sec. 3), one can show that Game 6 and Game 7 are indistinguishable due to the selective security of WPRF.

In Game 7, the encryption key K becomes independent of the challenge attribute. Therefore, $\mathsf{SKE.Enc}(\mathsf{K}, m_0)$ is indistinguishable from $\mathsf{SKE.Enc}(\mathsf{K}, m_1)$ by the LP-CCA security of SKE (Remark 1). This completes the proof.

A.2 Proof of Th. 4

Proof. The proof is similar to that of Th. 3 with few changes. Firstly, we replace the SKE by the semantically secure MFHE throughout the proof of Th. 3. Secondly, observe that the semantic security of MFHE is sufficient as we consider $\mathsf{CCA1}$ security for which $\mathcal{A}$ is not allowed to query the decryption oracle after the challenge query. More specifically, the secret-key sk_{v^*}, associated with the challenge ciphertext, is not needed for any decryption oracle used in the proof. We omit the lines 5 and 6 from both the oracles $O_{\mathsf{D},\mathsf{K}}$ and $O_{\mathsf{D}^*,\mathsf{K}}$, and rename them by $O_{\widetilde{\mathsf{D}}}$ and $O_{\widetilde{\mathsf{D}^*}}$ respectively. Finally, in $\mathsf{Game}\ 7$ we select y^* uniformly at random instead of setting it as $y^* \leftarrow \mathsf{WPRF.Eval}(\mathsf{fk}, (x^*, v^*))$. Thus, the MFHE key pair $(\mathsf{pk}_{v^*}, \mathsf{sk}_{v^*}) \leftarrow \mathsf{MFHE.KeyGen}(\mathsf{params}; y^*)$ is independent of the challenge attribute. Hence, the semantic security of MFHE ensures that the ciphertext distributions $(x^*, \mathsf{MFHE.Enc}(\mathsf{pk}_{v^*}, 0), v^*, \mathsf{pk}_{v^*})$ and $(x^*, \mathsf{MFHE.Enc}(\mathsf{pk}_{v^*}, 1), v^*, \mathsf{pk}_{v^*})$ are indistinguishable which completes the proof.