Multi-Identity IBFHE and Multi-Attribute ABFHE in the Standard Model

Xuecheng Ma[1,2] and Dongdai Lin[1,2](✉)

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China
{maxuecheng,ddlin}@iie.ac.cn

Abstract. The notion of multi-identity IBFHE is an extension of identity based fully homomorphic (IBFHE) encryption. In 2015, Clear and McGoldrick (CRYPTO 2015) proposed a multi-identity IBFHE scheme that is selectively secure in the random oracle model under the hardness of Learning with Errors (LWE). At TCC 2016, Brakerski et al. presented multi-target ABFHE in the random oracle where the evaluator should know the target policy. In this paper, we present a multi-identity IBFHE scheme and a multi-attribute ABFHE scheme in the standard model. Our schemes can support evaluating circuits of unbounded depth but with one limitation: there is a bound N on the number of ciphertexts under the same identity or attribute involved in the computation. The bound N could be thought of as a bound on the number of independent senders. Our schemes allow N to be exponentially large so we do not think it is a limitation in practice. Our construction combines *fully* multi-key FHE and leveled *single-identity* IBFHE or *single-attribute* ABFHE, both of which have been realized from LWE, and therefore we can instantiate our construction that is secure under LWE. Moreover, our multi-attribute ABFHE is non-target where the public evaluator do not need to know the policy.

Key words: multi-identity, multi-attribute, homomorphic encryption, standard model

1 Introduction

Identity Based Encryption (IBE) is proposed in 1984 by Shamir [Sha84] which is a generalization of public key encryption where the public key of a user can be arbitrary string such as an email address, IP address or staff number, depending on the application. The first realizations of IBE are given by [SOK00,BF01] using groups equipped with bilinear maps. Subsequently, realizations from bilinear maps [BB04a,BB04b,Wat05,Wat09], from quadratic residues modulo composite [Coc01,BGH07], from lattices [GPV08,CHKP10,ABB10] and from the computational Diffie-Hellman assumption [DG17] have been proposed.

Attribute-based encryption (ABE)[3] [SW05,GPSW06] is a generalization of IBE that allows to implement access control. A (master) public key mpk is used for encryption, and users are associated to secret keys sk_f corresponding to policy functions $f : \{0,1\}^\ell \to \{0,1\}$. The encryption of a message μ is labeled with a public attribute $x \in \{0,1\}^\ell$, and can be decrypted using sk_f if and only if $f(x) = 0$. The security guarantee of ABE is collusion resistance: a coalition of users learns nothing about the plaintext message μ if none of their individual keys are authorized to decrypt the ciphertext. Goyal, Pandey, Sahai and Waters [GPSW06] constructed ABE for log-depth circuits using bilinear maps. Gorbunov, Vaikuntanathan and Wee [GVW13] presented the first ABE scheme where the policies can be arbitrary (a-priori bounded) polynomial circuits from LWE. Boneh et al. [BGG$^+$14] showed an ABE scheme improving the size of secret key.

Fully homomorphic encryption (FHE) is first presented in 1987 by Rivest, Adleman and Dertouzos [RAD78]. Then Gentry[Gen09a,Gen09b] proposed the first construction in a breakthrough work in 2009. Since then, there are some follow-up works [BV11,BGV12,Bra12,GSW13,AP14] for improving efficiency and security. In 2013, Gentry, Sahai and Waters [GSW13] proposed a FHE scheme without an evaluation key which makes it enable to compile IBE with some properties into identity based (leveled) fully homomorphic encryption(IBFHE). Their compiler can also be applied to ABE that yields an attribute based (leveled) fully homomorphic encryption. Clear and McGoldrick [CM14] make IBFHE and ABFHE bootstrappable by using programm obfuscation.

López-Alt, Tromer and Vaikuntanthan [LTV12] considered an extension of homomorphic encryption into the multi-key setting, where it is possible to compute on encrypted messages even if they were not encrypted using the same key. In multi-key FHE(MKFHE), a public evaluator takes ciphertexts encrypted under different keys, and evaluates arbitrary (maybe with bounded depth) functions on them. The resulting ciphertext can then be decrypted using the collection of keys of all parties involved in the computation. Note that the security of the encryption scheme compels that all secret keys need to be used for decryption. [LTV12] constructed an on-the-fly multiparty computation (MPC) protocol by applying multi-key FHE. The next step forward was by Clear and McGoldrick [CM15] who proposed a multi-key FHE in the standard model and a multi-identity IBFHE in the random oracle. Note that the compiler in [GSW13] can only yield a *single-identity* IBFHE. As a stepping stone, they were able to construct a multi-key FHE scheme based on the hardness of the learning with errors (LWE) problem [Reg05,Reg09], which is related to the hardness of certain short vector problems (such as GapSVP, SIVP) in worst case lattices.

The first multi-identity IBFHE proposed by [CM15] is a leveled multi-identity IBFHE in the random oracle. [CM16] constructed a *fully* multi-identity IBFHE scheme that supports unbounded-depth circuits with bounded inputs by combining a MKFHE and multi-identity IBFHE. A natural question is that *Can*

[3] There are other variants such as ciphertext-policy ABE, but we focus on key-policy ABE here.

we construct a multi-identity IBFHE supporting unbounded-depth circuits in the standard model on the standard assumption?

Brakerski et al. [BCTW16] proposed a leveled multi-target ABFHE in the random model where the evaluator should know the target policy. Hiromasa and Kawai [HK17] extended leveled multi-target ABFHE to dynamic homomorphic evaluation. But these multi-target ABFHE schemes are all in the random oracle model. *Can we construct a multi-attribute ABFHE supporting unbounded-depth circuits in the standard model on the standard assumption?*

1.1 Our Contributions

We propose constructions of multi-identity IBFHE and multi-attribute ABFHE in the standard model. Our schemes support unbounded evaluation circuit depth but with one limitation: the number of ciphertexts joining the function computation under the same identity or attribute is bounded but can be exponential. Our construction combines a *fully* MKFHE and *single-identity* IBFHE or *single-attribute* ABFHE. There are instantiations of *single-identity* IBFHE, *fully* MKFHE [CM15,BP16,PS16,CZW17] and *single-attribute* ABFHE [GSW13] from LWE, so our construction can be instantiated from LWE. In order to construct a multi-identity IBFHE or multi-attribute ABFHE, Clear and McGoldrick [CM16] combine a fully MKFHE and leveled multi-identity IBFHE or multi-attribute ABFHE. So their proposal is not in the standard model without a multi-identity IBFHE or multi-attribute ABFHE in the standard model. In fact, their purpose is to make the multi-identity IBFHE support unbounded evaluation circuit depth while our goal is to construct a multi-identity IBFHE or multi-attribute ABFHE that supports *unbounded* evaluation circuit depth in the *standard model*. Brakerski et al. [BCTW16] proposed a multi-target ABFHE in the random oracle where the policy should be known to the evaluator while ours is *non-target* in the *standard model*.

1.2 Our Construction

Our construction is combining a MKFHE and a *single-identity* IBFHE or *single-attribute* ABFHE. The constructions of multi-identity IBFHE and multi-attribute ABFHE are similar, so we will show our high level idea of the construction of multi-identity IBFHE, the detail will be presented in Section 4. Our construction is similar to [CM16]. Both of our construction is using MKFHE to evaluate the circuit and then decrypts the evaluated MKFHE ciphertext by evaluation with the IBFHE encryptions of corresponding secret keys which makes the final resulting ciphertext compact. The difference here is that [CM16] decrypts the evaluated MKFHE cipherrtext completely which needs a multi-identity IBFHE while we just partially decrypts it which makes the final resulting IBFHE ciphertexts compact. It is the point that *single-identity* IBFHE works here.

We present the overview of our construction as follows: The setup algorithm generates params of MKFHE and (mpk, msk) of IBFHE by running their setup

algorithms respectively. The extract algorithm is the same as that of *single-identity* IBFHE. When encrypts a plaintext μ_i, the sender generates a pair of key (pk_i, sk_i) of MKFHE, and encrypts sk_i using encryption algorithm of IBFHE and then encrypts the plaintext using pk_i. When evaluates a function f, run the evaluation algorithm of MKFHE and obtain the MKFHE encryption of $f(\mu_1, ..., \mu_\ell)$, then partially decrypt the evaluated MKFHE ciphertext with IBFHE ciphertexts of the collection of secret keys corresponding the same identity. The resulting ciphertext is compact because the number of compact IBFHE ciphertexts in the final resulting ciphertexts are independent on the number of input ciphertexts and the size of the evaluation function. If the ciphertext is "fresh", we just obtain the secret key of MKFHE by decrypting IBFHE ciphertext and then decrypt the MKFHE ciphertext. If the ciphertext is evaluated, we can obtain the partial decryption by decrypting the IBFHE ciphertexts and finish the remaining decryption procedure of MKFHE.

1.3 Other Related Work

Clear and McGoldrick [CM15] extended the scheme of [GSW13] to the multi-identity setting and obtain a multi-identity IBFHE scheme that is selectively secure in the random oracle model under the hardness of Learning with Errors (LWE). Their scheme was simplified by Mukherjee and Wichs [MW16] who used multi-key FHE to construct a 2-round MPC protocols in the CRS model. Recently, Peikert and Shiehian [PS16] put forth a notion of *multi-hop* MKFHE, in which the result ciphertexts of homomorphic evaluations can be used in further homomorphic computations involving additional parties. Chen et al. [CZW17] then presented a compact *multi-hop* MKFHE which is based on Brakerski-Gentry-Vaikuntanathan (BGV) FHE scheme. Brakerski and Perlman [BP16] presented a similar notion called *fully dynamic* MKFHE that supports an unbounded number of homomorphic operations for an unbounded number of parties.Canetti et al. [CRRV17] show that CPA secure multi-identity IBFHE can be used to construct CCA1 secure homomorphic encryption.

2 Preliminaries

Let ℓ_q denote $\lfloor \log q \rfloor + 1$ and $\hat{m} = m \cdot \ell_q$. Let $a \in \mathbb{Z}_q^m$ be a vector of some dimension m over $\mathbb{Z}_q$ and $A \in \mathbb{Z}_q^{n \times m}$ be a matrix. $A[i]$ means the i-th row of A. We can see a vector as a matrix where $n = 1$. $\mathsf{BitDecomp}(a)$: We define an algorithm $\mathsf{BitDecomp}$ that takes as input a vector $a \in \mathbb{Z}_q^m$ and outputs an $\hat{m}$-dimensional vector $(a_{1,0}, ..., a_{1,\ell_q-1}, ..., a_{m,0}, ..., a_{m,\ell_q-1})$ where $a_{i,j}$ is the j-th bit in a_i's binary representation (ordered from least significant to most significant). $\mathsf{Binary}(A)$: It takes a matrix $A \in \mathbb{Z}_q^{n \times m}$ and outputs a $(n \cdot \hat{m})$-dimensional vector $(\mathsf{BitDecomp}(A[1]), ..., \mathsf{BitDecomp}(A[n]))$.

Definition 1 ([BHHO08]) *A public key encryption scheme* PKE *is said to be* weakly circular secure *if it is secure even against an adversary who gets encryptions of the bits of the secret key.*

2.1 Multi-Identity IBFHE

A Multi-Identity IBFHE scheme is defined with respect to a message space $\mathcal{M}$, an identity space $\mathcal{I}$, a class of circuits $\mathbb{C} \subset \mathcal{M}^* \to \mathcal{M}$ and ciphertext space C. A Multi-identity IBFHE scheme is a tuple of *ppt* algorithms (Setup, KeyGen, Encrypt, Decrypt, Eval) defined as follows:

- $\mathsf{Setup}(1^\lambda)$: On input (in unary) a security parameter λ, generate public parameters MPK and a master secret key MSK. Output (MPK, MSK).
- KeyGen(MSK, id): On input master secret key MSK and an identity id: derive and output a secret key $\mathsf{sk}_{\mathsf{id}}$ for identity id.
- Encrypt(MPK, id, μ): On input public parameters MPK, an identity id, and a message $\mu \in \mathcal{M}$, output a ciphertext $c \in C$ that encrypts μ under identity id.
- $\mathsf{Decrypt}(\mathsf{sk}_{\mathsf{id}_1}, ..., \mathsf{sk}_{\mathsf{id}_n}, c)$: On input n secret keys $\mathsf{sk}_{\mathsf{id}_1}, ..., \mathsf{sk}_{\mathsf{id}_n}$ for (resp.) identities $\mathsf{id}_1, ..., \mathsf{id}_n$ and a ciphertext $c \in C$, output $\mu \in \mathcal{M}$ if c is a valid encryption under identities $\mathsf{id}_1, ..., \mathsf{id}_n$; output a failure symbol $\perp$ otherwise.
- Eval(MPK, C, $c_1, ..., c_\ell$): On input public parameters MPK, a circuit $\mathsf{C} \in \mathbb{C}$ and ciphertexts $c_1, ..., c_\ell \in C$, output an evaluated ciphertext $\hat{c} \in C$.

For all choices of $\mathsf{Setup}(1^\lambda) \to (\mathsf{MPK}, \mathsf{MSK})$,$\mathsf{id}_1, ..., \mathsf{id}_n, j_1, ..., j_\ell \in [n]$, $c_i = \mathsf{Encrypt}(\mathsf{MPK}, \mathsf{id}_{j_i}, \mu_i)(\mu_i \in \mathcal{M})$,$\mathsf{C} : \mathcal{M}^\ell \to \mathcal{M}$,$\hat{c} = \mathsf{Eval}(\mathsf{MPK}, \mathsf{C}, c_1, ..., c_\ell)$

- **Correctness.**

$$\mathsf{Decrypt}(\mathsf{sk}_{\mathsf{id}_1}, ..., \mathsf{sk}_{\mathsf{id}_n}, \hat{c}) = \mathsf{C}(\mu_0, ...\mu_\ell)$$

- **Compactness.**

$$|\hat{c}| \leq \mathsf{poly}(\lambda, n)$$

  where n is the number of distinctive identities.
- **Security.** The security of multi-identity IBFHE is the same with the security of IBE.

3 Building Blocks from Previous Works

3.1 Fully Multi-key FHE

A homomorphic encryption scheme is multi-key if it can evaluate circuits on ciphertexts encrypted under different public keys. It is called leveled MKFHE if its setup algorithm needs to take a supported evaluation circuit depth as a input. Any leveled MKFHE [CM15,PS16,CZW17] with additional weakly circular security assumption can be converted into a *fully* MKFHE scheme. To decrypt an evaluated ciphertext, the decryption algorithm uses the secret keys of all parties involved in the computation. In fact, we need the MKFHE with threshold decryption property. We will define a generalized threshold decryption property

called *subset threshold decryption* and show that we can realize it by modifying existing threshold decryption multi-key FHE.

A multi-key homomorphic encryption scheme MKFHE = ($\mathsf{MKFHE.Setup}$, $\mathsf{MKFHE.Keygen}$, $\mathsf{MKFHE.Encrypt}$, $\mathsf{MKFHE.Decrypt}$, $\mathsf{MKFHE.Eval}$) is a 5-tuple of *ppt* algorithms as follows:

- **Setup** $\mathsf{params} \leftarrow \mathsf{MKFHE.Setup}(1^\lambda)$: Takes the security parameter as input and outputs the public parametrization params of the system.
- **Key generation** $(pk, sk) \leftarrow \mathsf{MKFHE.Keygen}(\mathsf{params})$: Outputs a public encryption key pk and a secret decryption key sk.
- **Encryption** $c \leftarrow \mathsf{MKFHE.Encrypt}(pk, \mu)$: Using the public key pk, encrypts a single bit message $\mu \in \{0,1\}$ into a ciphertext c.
- **Decryption** $\mu \leftarrow \mathsf{MKFHE.Decrypt}((sk_1, ..., sk_{\hat{N}}), c)$: Using the sequence of secret keys $(sk_1, ..., sk_{\hat{N}})$, decrypts a ciphertext c to recover the message $\mu \in \{0,1\}$.
- **Evaluation** $\hat{c} \leftarrow \mathsf{MKFHE.Eval}(\mathsf{C}, (c_1, ..., c_\ell), (pk_1, ..., pk_{\hat{N}}))$: Using the sequence of public keys $(pk_1, ..., pk_{\hat{N}})$, applies a circuit $\mathsf{C} : \{0,1\}^\ell \to \{0,1\}$ to $(c_1, ..., c_\ell)$, where each ciphertext c_j is evaluated under a sequence of public keys $V_j \subset \{pk_1, ..., pk_{\hat{N}}\}$ (we assume that V_j is implicit in c_j). Upon termination,outputs a ciphertext $\hat{c}$.

  Remark 1 *In multi-key GSW scheme, there is a* Expand *algorithm which takes a ciphertext* c_j *under* pk_j *and* V_j *where* $V_j \subset \{pk_1, ..., pk_{\hat{N}}\}$ *and* $pk_j \in V_j$ *as inputs and outputs a expanded ciphertext* $\hat{c}_j$ *which is the encryption of the same plaintext encrypted by* c_j *under all of public keys in* V_j

Definition 2 (fully multi-key FHE) *A scheme* MKFHE *is fully multi-key FHE, if the following holds. Let* $\hat{N} = \hat{N}_\lambda$ *be any polynomial in the security parameter,* $\mathsf{C} = \mathsf{C}_\lambda$ *be a sequence of circuits. For all* $\mathsf{params} \leftarrow \mathsf{MKFHE.Setup}(1^\lambda)$, $(pk_i, sk_i) \leftarrow \mathsf{MKFHE.Keygen}(\mathsf{params})(i \in [\hat{N}])$, $\mu_j \in \{0,1\}_{j\in[\ell]}$. $\mathsf{MKFHE.Decrypt}(c_j, sk_{i_j}) = \mu_j$ *where* $\{sk_{i_j} \in \{sk_1, ..., sk_{\hat{N}}\}\}_{j\in[\ell]}$, $\hat{c} \leftarrow \mathsf{MKFHE.Eval}(\mathsf{C}, (c_1, ..., c_\ell), (pk_1, ..., pk_{\hat{N}}))$.

- ***Correctness.***

$$\mathsf{C}(\mu_0, ..., \mu_\ell) = \mathsf{MKFHE.Decrypt}(\hat{c})$$

- ***Compactness.***

$$|\hat{c}| \leq \mathsf{poly}(\lambda, \hat{N})$$

  where $\hat{N}$ *is the number of distinctive public keys whose corresponding ciphertexts joining the computation.*

Semantic Security. The definition of $\mathsf{IND\text{-}CPA}$ security for MKFHE is the same as that for standard public-key encryption. It works for the multi-key setting because if any adversary $\mathcal{A}$ who can distinguish expanded (possibly evaluated)

ciphertexts of two equal-length plaintext can be used to distinguish two equal-length plaintext encryptions of public-key encryption. There exists a simulator $\mathcal{B}$ that can break IND-CPA security of PKE [4] with the help of $\mathcal{A}$. The challenger generates $(pk_1, sk_1) \leftarrow$ MKFHE.KeyGen(params) (we suppose the params here is common information). $\mathcal{B}$ receives pk_1 from the challenger and sends it to $\mathcal{A}$. $\mathcal{A}$ generates $\hat{N}-1$ pairs of keys $\{(pk_i, sk_i) \leftarrow \mathsf{MKFHE.KeyGen}(\mathsf{params})\}_{i\in\{2,...,\hat{N}\}}$ and sends $pk_2, ...pk_{\hat{N}}$ and two equal-length messages (μ_0, μ_1) to $\mathcal{B}$. $\mathcal{B}$ forwards (μ_0, μ_1) to the challenger and obtains the challenge ciphertext c from it. $\mathcal{B}$ can expand c into a ciphertext $\hat{c}$ under $pk_1, ..., pk_{\tilde{N}}$ and sends it to $\mathcal{A}$. $\mathcal{B}$ just forwards $\mathcal{A}$'s guess. If $\mathcal{A}$ can guess right with probability $\frac{1}{2}+\xi$, then the advantage of $\mathcal{B}$ breaks IND-CPA of PKE is ξ. The reason we define the security of multi-identity IBFHE and multi-attribute ABFHE as the security of IBE and ABE respectively is similar.

We now define a multi-key FHE which supports a one-round generalized threshold distributed decryption protocol called subset threshold decryption. Such a protocol consists of two components: (1) given an expanded ciphertext (possibly evaluated) c each subset can compute a partial decryption using its corresponding secret keys, (2) there is a way to combine the partial decryptions computed by each subset to recover the plaintext. It is easy to know that threshold decryption is just a special case that there is only one element in each subset.

Definition 3 *A* Subset Threshold multi-key FHE scheme *is a MKFHE scheme with two additional algorithms* MFHE.SubsetDec, MFHE.CombineDec *described as follows:*

- $\mathsf{h_i} \leftarrow \mathsf{MKFHE.SubsetDec}(c, (pk_1, ..., pk_{\hat{N}}), I_{i_1}, ...I_{i_{|\mathcal{T}_i|}}, sk_{I_{i_1}}, ..., sk_{I_{i_{|\mathcal{T}_i|}}})$*: On input an expanded ciphertext(possibly evaluated) under a sequence of $\hat{N}$ public keys and corresponding secret keys $sk_{i_1}, ..., sk_{i_{|\mathcal{T}_i|}}$ of the i-th index subset $\mathcal{T}_i$ and outputs a partial decryption* $\mathsf{h_i}$*. Here* $\mathcal{T}_i = \{I_{i_1}, ..., I_{i_{t_i}}\}$ *where* $I_{i_j} \in [\hat{N}]$*,* $t_i = |\mathcal{T}_i|$*.*
- $\mu \leftarrow \mathsf{MKFHE.CombineDec}(\mathsf{h_1}, ..., \mathsf{h_n})$*: On input n partial decryption outputs the plaintext* μ*.*

Along with the properties of multi-key FHE we require the scheme to satisfy the correctness and security.

Correctness. *Let* $\mathsf{params} \leftarrow \mathsf{MKFHE.Setup}(1^\lambda)$*. For any sequences of* $\hat{N}$ *correctly generated key pairs* $\{(pk_i, sk_i) \leftarrow \mathsf{MKFHE.Keygen}(\mathsf{params})\}_{i\in[\hat{N}]}$ *and any* ℓ*-tuple of messages* $(\mu_1, ..., \mu_\ell)$*. For set of indices* $\mathcal{T} = \{1, ..., \hat{N}\}$ *and any n subsets of* $\mathcal{T}$ $\mathcal{T}_1, ...\mathcal{T}_n$*, where* $\mathcal{T}_i \cap \mathcal{T}_j = \emptyset$ $(i \neq j)$ *and* $\mathcal{T}_1 \cup ... \cup \mathcal{T}_n = \mathcal{T}$*. We denote* $\mathcal{T}_i$ *as* $\{I_{i_1}, ..., I_{i_{t_i}}\}$*. Let* $R : [\ell] \rightarrow [\hat{N}]$ *denote a function from indices of plaintexts to indices of public keys and* $\{c_k \leftarrow \mathsf{Encrypt}(pk_{R(k)}, \mu_k)\}_{k\in[\ell]}$ *be encryptions of the messages* μ_k *under the* $R(k)$*-th public key. Let* C *be any (boolean) circuit and let*

[4] The PKE is not a general PKE here. Its setup, encryption, decryption algorithms are the same as the MKFHE scheme.

$\hat{c} := Eval(\mathsf{C}, (c_1, ..., c_\ell)$ *be the evaluated ciphertext. The below equation should hold with probability 1.*

$$\mathsf{MKFHE.CombineDec}(\mathsf{h_1}, ..., \mathsf{h_n}) = \mathsf{C}(\mu_1, ..., \mu_\ell)$$

$\{\mathsf{h_i} \leftarrow \mathsf{MKFHE.SubsetDec}(c, pk_1, ..., pk_{\hat{N}}, I_{i_1}, ..., I_{i_{|\mathcal{T}_i|}}, sk_{I_{i_1}}, ..., sk_{I_{i_{|\mathcal{T}_i|}}})\}_{i\in[n]}$ *are partial decryptions and* $\{\mathcal{T}_i = \{I_{i_1}, ..., I_{i_{t_i}}\}\}_{i\in[n]}$ *in above equation.*
Security. *The semantic security of* MKFHE *with subset threshold decryption should hold. It is trivial because the* IND-CPA *security does not dependent on decryption algorithm.*

We will show that we can easily convert the threshold decryption of multi-key GSW into our subset threshold decryption. In fact, threshold decryption defined in [MW16] also has two similar algorithms[5] PartDec and FinDec where PartDec takes the evaluated ciphertext c, all parties' public keys $(pk_1, ..., pk_{\hat{N}})$ and one party's secret key sk_i and outputs the partial decryption p_i, and FinDec takes all partial decryptions $(p_1, ..., p_{\hat{N}})$ as inputs and outputs the plaintext μ. We observe that the FinDec algorithm of GSW-type scheme is $\sum_{i=1}^{\hat{N}} p_i$. So we can instantiate our SubsetDec and CombineDec algorithms as follows:
$\mathsf{SubsetDec}(c, pk_1, ..., pk_{\hat{N}}, I_{i_1}, ..., I_{i_{|\mathcal{T}_i|}}, sk_{I_{i_1}}, ..., sk_{I_{i_{|\mathcal{T}_i|}}})$:

$$\{p_{i_j} \leftarrow \mathsf{PartDec}(c, pk_1, ..., pk_{\hat{N}}, I_{i_j}, sk_{I_{i_j}})\}_{j\in[|\mathcal{T}_i|]},\ \mathsf{h_i} = \sum_{j=1}^{|\mathcal{T}_i|} p_{i_j}$$

$\mathsf{CombineDec}(\mathsf{h_1}, ..., \mathsf{h_n})$: $\sum_{i=1}^{n} \mathsf{h_i}$

We refer to $\overline{\mathsf{SubsetDec}}[\mathcal{T}_i]$ as the circuit that SubsetDec algorithm takes $\mathcal{T}_i$ as the indices components of inputs.

3.2 Leveled IBFHE

A leveled IBFHE scheme is defined with respect to a message space $\mathcal{M}$, an identity space $\mathcal{I}$, a class of circuits $\mathbb{C} \subset \mathcal{M}^* \to \mathcal{M}$ and ciphertext space C. An IBFHE scheme is a tuple of *ppt* algorithms (Setup, KeyGen, Encrypt, Decrypt, Eval) defined as follows:

- $\mathsf{Setup}(1^\lambda, L)$: On input (in unary) a security parameter λ and the bounded evaluation circuit depth L supported, generate public parameters MPK and a master secret key MSK. Output (MPK, MSK).
- KeyGen(MSK, id): On input master secret key MSK and an identity id: derive and output a secret key $\mathsf{sk_{id}}$ for identity id.
- Encrypt(MPK, id, μ): On input public parameters MPK, an identity id, and a message $\mu \in \mathcal{M}$, output a ciphertext $c \in C$ that encrypts μ under identity id.

[5] More details of the two algorithms can be found in [MW16]

- $\mathsf{Decrypt}(\mathsf{sk}_{\mathsf{id}}, c)$: On input secret key $\mathsf{sk}_{\mathsf{id}}$ for (resp.) identity id and a ciphertext $c \in \mathcal{C}$, output $\mu \in \mathcal{M}$ if c is a valid encryption under identities id; output a failure symbol $\perp$ otherwise.
- $\mathsf{Eval}(\mathsf{MPK}, \mathsf{C}, \mathsf{id}, c_1, ..., c_\ell)$: On input public parameters MPK, a circuit $\mathsf{C} \in \mathbb{C}$ and ciphertexts $c_1, ..., c_\ell \in C$ under id, output an evaluated ciphertext $\hat{c} \in C$ under id.

For all choices of $\mathsf{Setup}(1^\lambda, L) \rightarrow (\mathsf{MPK}, \mathsf{MSK})$, $c_i = \mathsf{Encrypt}(\mathsf{MPK}, \mathsf{id}, \mu_i)(\mu_i \in \mathcal{M})$,$\mathsf{C} : \mathcal{M}^* \rightarrow \mathcal{M}$ whose depth is less than L, $\hat{c} = \mathsf{Eval}(\mathsf{MPK}, \mathsf{C}, c_1, ..., c_\ell)$

- **Correctness.**

$$\mathsf{Decrypt}(\mathsf{sk}_{\mathsf{id}}, \hat{c}) = \mathsf{C}(\mu_0, ...\mu_\ell)$$

- **Compactness.**

$$|\hat{c}| \leq \mathsf{poly}(\lambda, L)$$

4 Multi-Identity IBFHE

4.1 Construction

We combine a multi-key FHE and *single-identity* IBFHE to construct our multi-identity IBFHE scheme. Setup algorithm outputs public parameters and master secret key of IBFHE and params of MKFHE by running their setup algorithms respectively. When encrypt a plaitext $\mu \in \{0, 1\}$, the sender generates $(pk, sk) \leftarrow \mathsf{MKFHE.KeyGen}(\mathsf{params})$, then encrypts sk under id and μ under pk. The evaluator evaluates the circuit on MKFHE ciphertexts and obtain an evaluated ciphertext $\hat{c}$. Then it evaluates with the leveled IBFHE scheme the partial decryption circuit $\overline{\mathsf{SubsetDec}}[\mathcal{T}_j]$ for all $j \in [n]$ where $\mathcal{T}_j$ is the set of indices of corresponding public keys for id_j. The number of (compact) evaluated IBFHE ciphertext is independent on the number of senders which makes the whole resulting ciphertext compact. Receivers can obtain partial decryption of the evaluated MKFHE ciphertext by decrypting the IBFHE ciphertext under its identity. Then they can jointly decrypt the evaluated MKFHE ciphertext. Our construction is *fully* multi-identity IBFHE with additional weakly circular security where we do not need to take circuit depth as input in the Setup algorithm. In order to compute a function in our construction we will assign every plaintexts, every pair of (public and secret) keys of MKFHE and identities of IBFHE indices. Let the pair of keys and plaintext share the same index because each public key of MKFHE only encrypts one plaintext. For example, if we use pk to encrypts μ_i, we denote pk as pk_i and sk as sk_i. We can use lexicographic order of identities as their indices. Suppose there are ℓ plaintexts and n different identities involved in the computation, we can define a function $\hat{R} : [\ell] \rightarrow [n]$ where $\hat{R}(i) = j$ if pk_i is generated in the encryption process for id_j. Set the preimages of j as $\mathcal{T}_j = \{I_{(\mathsf{id}_j,1)}, ..., I_{(\mathsf{id}_j,t_j)}\}$ where $I_{(\mathsf{id}_j,1)}, ..., I_{(\mathsf{id}_j,t_j)}$ are indices of the ciphertexts for the same identity id_j.

- $\mathsf{Setup}(1^\lambda, N)$: Take the security parameter and the bound of number of ciphertexts under the same identity that the system can tolerate. Compute $\mathsf{params} \leftarrow \mathsf{MKFHE.Setup}(1^\lambda)$, $(\mathsf{MPK}', \mathsf{MSK}') \leftarrow \mathsf{IBFHE.Setup}(1^\lambda, L)$, where $L = \tau(N, \lambda)$ is the depth of the decryption circuit of MKFHE for parameters[6] λ and N. Output $(\mathsf{MPK}, \mathsf{MSK}) = ((\mathsf{MPK}', \mathsf{params}), \mathsf{MSK}')$.
- $\mathsf{KeyGen}(\mathsf{MSK}, \mathsf{id})$: This algorithm is the same as IBFHE. Just output $\mathsf{sk}_{\mathsf{id}} = \mathsf{IBFHE.KeyGen}(\mathsf{MSK}, \mathsf{id})$.
- $\mathsf{Encrypt}(\mathsf{MPK}, \mathsf{id}, \mu \in \{0,1\})$: Run $(pk, sk) \leftarrow \mathsf{MKFHE.KeyGen}(\mathsf{params})$. Compute $c' \leftarrow \mathsf{MKFHE.Encrypt}(pk, \mu)$, $\phi \leftarrow \mathsf{IBFHE.Encrypt}(\mathsf{MPK}', \mathsf{id}, sk)$. Output $c = (\mathsf{type} := 0, \mathsf{enc} := (c', \phi, \mathsf{id}, pk))$.
- $\mathsf{Eval}(\mathsf{MPK}, \mathsf{C}, c_1, ..., c_\ell)$: The ciphertexts are assumed to be "fresh" ciphertexts generated with the encryption algorithm. In other words, their type components are all 0. Otherwise the evaluator outputs $\perp$. Parse c_i as $(\mathsf{type} := 0, \mathsf{enc} := (c_i', \phi_i, \mathsf{id}_{\hat{R}(i)}, pk_i)$. Firstly, evaluate the circuit on MKFHE ciphertexts. Compute $\hat{c} = \mathsf{MKFHE.Eval}(\mathsf{C}, (c_1', ..., c_\ell'), (pk_1, ..., pk_\ell))$. For all $j \in [n]$, proceed as following two steps. Step 1: encrypt the evaluated MKFHE ciphertext under id_j. Let $\hat{c}_{\mathsf{bin}} = \mathsf{Binary}(\hat{c})$. Compute $\{\bar{c}_i \leftarrow \mathsf{IBFHE.Encrypt}(\mathsf{MPK}', \mathsf{id}_j, \hat{c}_{\mathsf{bin}}[i])\}_{i \in [|\hat{c}_{\mathsf{bin}}|]}$ the IBFHE encryption of every bit of evaluated ciphertext $\hat{c}$. Step 2: evaluate partial decryption circuit $\overline{\mathsf{SubsetDec}}[\mathcal{T}_j]$ on the ciphertexts $(\{\phi_{I_{(\mathsf{id}_j,k)}}\}_{k \in |\mathcal{T}_j|}, \{\bar{c}_i\}_{i \in [|\hat{c}_{\mathsf{bin}}|]})$ and obtain the IBFHE encryption c_{id_j} under id_j of the partial decryption of $\hat{c}$ where $c_{\mathsf{id}_j} = \mathsf{IBFHE.Eval}(\mathsf{MPK}', \overline{\mathsf{SubsetDec}}[\mathcal{T}_j], \mathsf{id}_j, \phi_{I_{(\mathsf{id}_j,1)}}, ..., \phi_{I_{(\mathsf{id}_j,|\mathcal{T}_j|)}}, \bar{c}_1, ...\bar{c}_{|\hat{c}_{\mathsf{bin}}|})$. Finally, outputs $c = (\mathsf{type} := 1, (c_{\mathsf{id}_1}, ..., c_{\mathsf{id}_n}))$
- $\mathsf{Decrypt}(\mathsf{sk}_{\mathsf{id}_1}, ..., \mathsf{sk}_{\mathsf{id}_n}, c)$: If c is a "fresh" ciphertext where $\mathsf{type} = 0$,we parse enc as $(c', \phi, \mathsf{id}, pk)$ and computes $sk = \mathsf{IBFHE.Decrypt}(\mathsf{sk}_{\mathsf{id}}, \phi)$. Computes $\mu = \mathsf{MKFHE.Decrypt}(c', sk)$ and outputs μ if $sk \neq \perp$. If c is an evaluated ciphertext (i.e. $\mathsf{type} = 1$), parse c as $(c_{\mathsf{id}_1}, ..., c_{\mathsf{id}_n})$, compute $\mathsf{h_i} = \mathsf{IBFHE.Decrypt}(\mathsf{sk}_{\mathsf{id}_i}, c_{\mathsf{id}_i})$ and outputs $\mu = \mathsf{MKFHE.CombineDec}(\mathsf{h_1}, ..., \mathsf{h_n})$. Otherwise output $\perp$.

Remark 2 *We can instantiate our* MKFHE *with the scheme of GSW-MKFHE [CM15,BP16,PS16] where its decryption circuit depth is* $O(log(N \cdot \lambda))$. *We set* N *to be a large value which dominates* λ, *so its decryption circuit depth is roughly* $O(logN)$. *For example, suppose we set* N *as* 2^{64}, *we need a leveled* IBFHE *that can evaluate* 64*-depth circuits.*

4.2 Main Results

Theorem 1 *Let* N *be a positive integer. Let* λ *be the security parameter. Let* n *be any polynomial in* λ. *Suppose there exists an* IND-CPA *secure subset threshold decryption* MKFHE *scheme that evaluates circuits of depth* d, *and its subset*

[6] In fact, if there exists a "pure" IBFHE, we don't need take N as input that makes our construction be a "pure" multi-key IBFHE.Unfortunately, [CM16] can only yield almost "pure" scheme which does not work here.

threshold decryption circuit depth is $\tau(N, \lambda)$. *Suppose that there exists an* IBFHE *scheme that can compactly evaluate circuits depth of* τ. *Then there exists a multi-identity* IBFHE *scheme supporting* n *identities that can compactly evaluate all* d*-depth boolean circuits in* $\{0,1\}^* \to \{0,1\}$ *with a limitation that the number of ciphertexts under the same identity is no more than* N.

Correctness

The construction is correct if MKFHE and leveled IBFHE are both correct. The decryption correctness of fresh ciphertext is guaranteed by the decryption correctness of MKFHE and IBFHE. If we set the parameters of IBFHE to support evaluation circuits depth larger than the depth of the decryption algorithm of MKFHE scheme, combining the evaluation and subset threshold decryption correctness of MKFHE and decryption correctness of IBFHE, the evaluated ciphertext can be decrypted correctly. So the correctness of our construction is guaranteed.

Compactness

If ciphertexts of IBFHE are compact, our construction is likewise compact. If ciphertexts of IBFHE is compact, $|c_{\mathsf{id}_j}| \leq \mathsf{poly}(\lambda, L)$, where L is a polynomial[7] in λ and larger than the depth of the decryption circuit of MKFHE. The evaluated ciphertext c is $n \cdot |\mathsf{h_i}|$ *compact* IBFHE ciphertexts where $|\mathsf{h}_i|$ is independent on the evaluated function and N. So we can conclude that $|c| \leq \mathsf{poly}(\lambda, n)$.

Security

Theorem 2 *Suppose that* MKFHE *is* IND-CPA *secure and single-identity* IBFHE *is* IND-X-CPA *secure,our construction is* IND-X-CPA *secure multi-identity* IBFHE *where* $\mathsf{X} \in \{\mathsf{Selective}, \mathsf{Adaptive}\}$.

Proof. We will prove the security by hybrid argument as follows.

Hybrid $\mathcal{H}_0$: This is identical to the IND-X-CPA game of multi-identity IBFHE.

Hybrid $\mathcal{H}_1$: Let id^* be the challenge identity the adversary sends. There is only one difference in the challenge ciphertext with $\mathcal{H}_0$. The challenger replaces the encryption of the secret key sk of the MKFHE (i.e. ϕ component of the challenge ciphertext) with $\phi \leftarrow \mathsf{IBFHE.Encrypt}(\mathsf{MPK}', \mathsf{id}^*, 0^{|sk|})$, where $0^{|sk|}$ is zeros whose length is the same as sk.

$\mathcal{H}_0$ and $\mathcal{H}_1$ is indistinguishable. In fact, if any *ppt* adversary $\mathcal{A}$ can distinguish them there exists a simulator $\mathcal{B}$ that can use $\mathcal{A}$ to break the IND-X-CPA of

[7] We see N as a constant here.

IBFHE. In the challenge phase, when $\mathcal{A}$ chooses a challenge identity id^*, $\mathcal{B}$ generates a pair of key for MKFHE i.e. it computes $\mathsf{params} \leftarrow \mathsf{MKFHE.Setup}(1^\lambda)$ and $(pk, sk) \leftarrow \mathsf{MKFHE.KeyGen}(\mathsf{params})$. Then $\mathcal{B}$ sends id^* and $(m_0 = sk, m_1 = 0^{|sk|})$ to its challenger. $\mathcal{B}$ obtains the challenge ciphertext from the challenger and set it as the ϕ component of its own challenge ciphertext c^* and then computes the remaining components of c^* via the encryption algorithm. $\mathcal{B}$ sends $\mathcal{A}$'s guess to its challenger. If ϕ is the encryption of sk, the view of $\mathcal{A}$ is identical to $\mathcal{H}_0$. If ϕ is the encryption of $0^{|sk|}$, the view of $\mathcal{A}$ is identical to $\mathcal{H}_1$. So the advantage of $\mathcal{B}$ breaks IND-X-CPA of IBFHE is equal to the advantage of $\mathcal{A}$ distinguishing $\mathcal{H}_0$ and $\mathcal{H}_1$. It is concluded that $\mathcal{H}_0$ and $\mathcal{H}_1$ are indistinguishable.

Hybrid $\mathcal{H}_2$: This is same as $\mathcal{H}_1$ except that the challenger dose not encrypt μ_0 or μ_1 sent by the adversary $\mathcal{A}$ in the challenge phase. It encrypts 0 instead. If $\mathcal{A}$ can distinguish $\mathcal{H}_1$ and $\mathcal{H}_2$ with a non-negligible advantage there exists a simulator $\mathcal{B}$ can break the IND-CPA security of MKFHE. $\mathcal{B}$ sends the public key pk obtained from its challenger to $\mathcal{A}$. $\mathcal{A}$ chooses two plaintext $\mu_0 \in \{0, 1\}$ and $\mu_1 \in \{0, 1\}$ as the challenge plaintext pair to $\mathcal{B}$. $\mathcal{B}$ randomly choose a bit b and sends (μ_b,0) to its challenger. $\mathcal{B}$ obtains a ciphertext c' from its challenger and set it as the MKFHE component of its challenge ciphertext c^* answered to $\mathcal{A}$. $\mathcal{B}$ computes the remaining components of c^*. $\mathcal{B}$ outputs 0 if $\mathcal{A}$'s guess is $\mathcal{H}_1$, and 1 otherwise. If c' encrypts μ_b, the view of $\mathcal{A}$ is identical to $\mathcal{H}_1$. If c' encrypts 0, the view of $\mathcal{A}$ is identical to $\mathcal{H}_2$. So $\mathcal{H}_1$ is indistinguishable from $\mathcal{H}_2$ if MKFHE is IND-CPA secure. In $\mathcal{H}_2$, the advantage of the adversary is zero because there are no information about the bit the challenger chooses in the challenge ciphertext.

Optimization

In fact, we can choose an integer ω in the setup stage and encrypt ω bits under one public key of MKFHE. We need an additional hybrid argument of multiple encryptions of MKFHE in the proof of security.

5 Multi-Attribute ABFHE

In this section, we will show the construction of multi-attribute ABFHE. The construction and proof are similar to those of multi-identity IBFHE. We give the proof in full version. Let $\mathcal{X}$ denotes attribute space and $\mathcal{F}$ denotes policy space. Suppose there are ℓ plaintexts and n different attributes involved in the computation, we can define a function $R' : [\ell] \to [n]$ where $R'(i) = j$ if pk_i is generated in the encryption process for x_j. Set the preimages of j as $\mathcal{T}_j = \{I_{(x_j,1)}, ..., I_{(x_j,t_j)}\}$ where $I_{(x_j,1)}, ..., I_{(x_j,t_j)}$ are indices of the ciphertexts for the same identity x_j.

- $\mathsf{Setup}(1^\lambda, N)$: Take the security parameter and the bound of number of ciphertexts for the same attribute that the system can tolerate. Compute $\mathsf{params} \leftarrow \mathsf{MKFHE.Setup}(1^\lambda)$, $(\mathsf{MPK}', \mathsf{MSK}') \leftarrow \mathsf{ABFHE.Setup}(1^\lambda, L)$, where

L is the depth of the decryption circuit of MKFHE for parameters, λ and N. Output $(\mathsf{MPK},\mathsf{MSK}) = ((\mathsf{MPK}',\mathsf{params}),\mathsf{MSK}')$.

- $\mathsf{KeyGen}(\mathsf{MSK},\ f \in \mathcal{F})$: This algorithm is the same as ABFHE. Just output $\mathsf{sk}_f = \mathsf{IBFHE.KeyGen}(\mathsf{MSK},\ f)$.
- $\mathsf{Encrypt}(\mathsf{MPK},\ x \in \mathcal{X},\ \mu \in \{0,1\})$: $(pk, sk) \leftarrow \mathsf{MKFHE.KeyGen}(\mathsf{params})$. Compute $c' \leftarrow \mathsf{MKFHE.Enc}(pk, \mu)$, $\phi \leftarrow \mathsf{Encrypt}(\mathsf{MPK}',\ x,\ sk)$. Output $c = (\mathsf{type} := 0, \mathsf{enc} := (c', \phi, x, pk))$.
- $\mathsf{Eval}(\mathsf{MPK},\ \mathsf{C},\ c_1, ..., c_\ell)$: Firstly, the ciphertexts are assumed to be "fresh" ciphertexts generated with the encryption algorithm. In other words, their type components are all 0. Otherwise the evaluator outputs $\perp$. Parse c_i as $(\mathsf{type} := 0, \mathsf{enc} := (c'_i, \phi_i, x_{R'(i)}, pk_i)$. Compute $\hat{c} = \mathsf{MKFHE.Eval}(\mathsf{C},(c'_1, ..., c'_\ell), (pk_1, ..., pk_\ell))$. Let $\hat{c}_{\mathsf{bin}} = \mathsf{Binary}(\hat{c})$. Compute $\{\bar{c}_i \leftarrow \mathsf{ABFHE.Encrypt}(\mathsf{MPK}', x_j, \hat{c}_{\mathsf{bin}}[i])\}_{i\in[|\hat{c}_{\mathsf{bin}}|]}$ the ABFHE encryption of every bit of evaluated ciphertext $\hat{c}$. Then evaluate partial decryption circuit $\overline{\mathsf{SubsetDec}}[\mathcal{T}_j]$ on ciphertexts $(\{\phi_{I_{x_j,k}}\}_{k\in|\mathcal{T}_j|}, \{\bar{c}_i\}_{i\in|c_{\mathsf{bin}}|})$ and obtain ABFHE encryption c_{x_j} under x_j of partial decryption of $\hat{c}$ where $c_{x_j} = \mathsf{ABFHE.Eval}(\mathsf{MPK}', \overline{\mathsf{SubsetDec}}[\mathcal{T}_\mathsf{j}], x_j, \phi_{I_{(x_j,1)}}, ..., \phi_{I_{(x_j,|\mathcal{T}_j|)}}, \bar{c}_1, ...\bar{c}_{|\hat{c}_{\mathsf{bin}}|})$ for all $j \in [n]$. Outputs $c = (\mathsf{type} :=1, (c_{x_1}, ..., c_{x_n}))$
- $\mathsf{Decrypt}(\mathsf{sk}_{\mathsf{f}_1}, ..., \mathsf{sk}_{\mathsf{f}_\mathsf{n}}, c)$: For simplicity[8], we suppose $f_i(x_i) = 0$ here. If c is a "fresh" ciphertext where $\mathsf{type} = 0$,we parse enc as (c', ϕ, x, pk) and compute $sk = \mathsf{ABFHE.Decrypt}(sk_{f_i}, \phi)$ where $x = x_i$. Compute $\mu = \mathsf{MKFHE.Decrypt}(c', sk)$ and output μ if $sk \neq \perp$. If c is an evaluated ciphertext (i.e. $\mathsf{type} = 1$), parse c as $(c_{x_1}, ..., c_{x_n})$, compute $\mathsf{h_i} = \mathsf{ABFHE.Decrypt}(sk_{f_i}, c_{x_i})$ and outputs $\mu = \mathsf{MKFHE.CombineDec}(\mathsf{h}_1, ..., \mathsf{h}_\mathsf{n})$. Otherwise output $\perp$.

References

[ABB10] Shweta Agrawal, Dan Boneh, and Xavier 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.

[AP14] Jacob Alperin-Sheriff and Chris Peikert. Faster bootstrapping with polynomial error. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, pages 297–314, 2014.

[BB04a] Dan Boneh and Xavier Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, pages 223–238, 2004.

[BB04b] Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, pages 443–459, 2004.

[8] It also works if sk_{f_i} can decrypt ciphertexts under many different attributes.

[BCTW16] Zvika Brakerski, David Cash, Rotem Tsabary, and Hoeteck Wee. Targeted homomorphic attribute-based encryption. In *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part II*, pages 330–360, 2016.

[BF01] Dan Boneh and Matthew K Franklin. Identity-based encryption from the weil pairing. *international cryptology conference*, 2001:213–229, 2001.

[BGG+14] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. 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 533–556, 2014.

[BGH07] Dan Boneh, Craig Gentry, and Michael Hamburg. Space-efficient identity based encryption without pairings. *IACR Cryptology ePrint Archive*, 2007:177, 2007.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*, pages 309–325, 2012.

[BHHO08] Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision diffie-hellman. In *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*, pages 108–125, 2008.

[BP16] Zvika Brakerski and Renen Perlman. Lattice-based fully dynamic multi-key FHE with short ciphertexts. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, pages 190–213, 2016.

[Bra12] Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. In *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, pages 868–886, 2012.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 97–106, 2011.

[CHKP10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris 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.

[CM14] Michael Clear and Ciaran McGoldrick. Bootstrappable identity-based fully homomorphic encryption. In *Cryptology and Network Security - 13th International Conference, CANS 2014, Heraklion, Crete, Greece, October 22-24, 2014. Proceedings*, pages 1–19, 2014.

[CM15] Michael Clear and Ciaran McGoldrick. Multi-identity and multi-key leveled FHE from learning with errors. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, pages 630–656, 2015.

[CM16] Michael Clear and Ciaran McGoldrick. Attribute-based fully homomorphic encryption with a bounded number of inputs. In *Progress in Cryptology - AFRICACRYPT 2016 - 8th International Conference on Cryptology in Africa, Fes, Morocco, April 13-15, 2016, Proceedings*, pages 307–324, 2016.

[Coc01] Clifford Cocks. An identity based encryption scheme based on quadratic residues. In Bahram Honary, editor, *Cryptography and Coding*, pages 360–363, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[CRRV17] Ran Canetti, Srinivasan Raghuraman, Silas Richelson, and Vinod Vaikuntanathan. Chosen-ciphertext secure fully homomorphic encryption. In *Public-Key Cryptography - PKC 2017 - 20th IACR International Conference on Practice and Theory in Public-Key Cryptography, Amsterdam, The Netherlands, March 28-31, 2017, Proceedings, Part II*, pages 213–240, 2017.

[CZW17] Long Chen, Zhenfeng Zhang, and Xueqing Wang. Batched multi-hop multi-key FHE from ring-lwe with compact ciphertext extension. In *Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part II*, pages 597–627, 2017.

[DG17] Nico Döttling and Sanjam Garg. Identity-based encryption from the diffie-hellman assumption. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part I*, pages 537–569, 2017.

[Gen09a] Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. `crypto.stanford.edu/craig`.

[Gen09b] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 169–178, 2009.

[GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, Ioctober 30 - November 3, 2006*, pages 89–98, 2006.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 197–206, 2008.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pages 75–92, 2013.

[GVW13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 545–554, 2013.

[HK17] Ryo Hiromasa and Yutaka Kawai. Fully dynamic multi target homomorphic attribute-based encryption. *IACR Cryptology ePrint Archive*, 2017:373, 2017.

[LTV12] Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *Proceedings of the 44th Symposium on Theory of Computing*

Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012, pages 1219–1234, 2012.

[MW16] Pratyay Mukherjee and Daniel Wichs. Two round multiparty computation via multi-key FHE. In *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, pages 735–763, 2016.

[PS16] Chris Peikert and Sina Shiehian. Multi-key FHE from lwe, revisited. In *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part II*, pages 217–238, 2016.

[RAD78] Ronald L Rivest, Len Adleman, and Michael L Dertouzos. On data banks and privacy homomorphisms. *Foundations of Secure Computation*, pages 169–179, 1978.

[Reg05] Oded 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.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009.

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, California, USA, August 19-22, 1984, Proceedings*, pages 47–53, 1984.

[SOK00] R Sakai, K Ohgishi, and M Kasahara. Cryptosystem based on pairings. 01 2000.

[SW05] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, pages 457–473, 2005.

[Wat05] Brent Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, pages 114–127, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[Wat09] Brent Waters. Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009*, pages 619–636, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.