IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 8 March 2023, accepted 27 March 2023, date of publication 5 April 2023, date of current version 12 April 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3264497*

RESEARCH ARTICLE

# Leveled Fully Homomorphic Signcryption From Lattices

**XIAODAN JIN[1,2], FUQUN WANG[1,2], RENJUN ZHANG[1,2], BIN LIAN[3], AND KEFEI CHEN[1,2], (Member, IEEE)**

[1]School of Mathematics, Hangzhou Normal University, Hangzhou 311121, China
[2]Key Laboratory of Cryptography of Zhejiang Province, Hangzhou Normal University, Hangzhou 311121, China
[3]School of Information Science and Engineering, NingboTech University, Ningbo 315199, China

Corresponding authors: Fuqun Wang (fqwang@hznu.edu.cn) and Kefei Chen (kfchen@hznu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61972124, Grant 11974096, and Grant 61972350; in part by the Research Foundation of Hangzhou Normal University under Grant 2020QDL016; in part by the Science and Technology Innovation 2025 Major Project of Ningbo under Grant 2021Z109; and in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LY23F020013.

**ABSTRACT** With the continuous and rapid development of Cloud Computing, Big Data and Internet of Things, it is extremely critical to protect data with homomorphism, privacy and integrity. For this, Rezaeibagha et al. proposed a new cryptographic primitive, called homomorphic signcryption. However, the current homomorphic signcryption schemes either only support linear computation or are built on non-standard assumption. Therefore, it is interesting to design a leveled fully homomorphic signcryption (FHSC) scheme from the standard assumption. In this work, we present a leveled FHSC scheme from lattices. For this, we exert classical sign-then-encrypt method and surmount the difficulty of homomorphic multiplicative evaluation in the way of encrypting every elements. Moreover, we prove its indistinguishability against chosen plaintext attacks (IND-CPA) and strong unforgeability (SUF) under hard problems of standard lattices.

**INDEX TERMS** Fully homomorphic signcryption, learning with errors, short integer solution.

## I. INTRODUCTION

With the integration and development of technologies such as Cloud Computing, Big Data and Internet of Things, there is an urgent requirement to ensure the privacy and integrity of data while providing the homomorphic evaluation. In 2009, Gentry [1] constructed a fully homomorphic encryption (FHE) scheme which can protect the data privacy but not the data integrity, while Gorbunov et al. [2] proposed a leveled fully homomorphic signature (FHS) scheme in 2015 which could provide the data integrity apart from the data privacy. There is an urgent demand to build a homomorphic scheme with privacy and integrity in the homomorphic computation setting.

For example, Smart Grid can perform functions such as intelligent power generation, balanced load, power distribution and dynamic price adjustment. Thus, it should ensure data privacy, integrity and homomorphism. In this setting, we can protect them in the way of signing-then-encrypting homomorphically, while supporting homomorphic evaluation.

For this, Rezaeibagha et al. [3] proposed a linearly homomorphic signcryption (HSC) scheme which demonstrates weak unforgeability (WUF) and indistinguishability against chosen plaintext attacks (IND-CPA) based on the Decisional Diffie-Hellman (DDH) assumption. Recently, Li et al. [4] constructed two leveled fully homomorphic signcryption (FHSC) schemes[1] with public plaintext-result checkability. One is in a public evaluation setting which employs FHS, indistinguishability obfuscation ($i\mathcal{O}$) and non-interactive zero knowledge proof. Another is in a private evaluation setting

The associate editor coordinating the review of this manuscript and approving it for publication was Kashif Saleem.

[1]To our best knowledge, currently the FHS schemes are all at most leveled. Their FHSC schemes are constructed from the FHS schemes and inherit the homomorphic capability of bottom FHS. Thus, the two FHSC schemes are leveled. Thus, we frequently omit "leveled" in this work.

which uses FHS, $i\mathcal{O}$ and mult-input functional encryption. Therefore, their schemes are not built from the standard assumption as $i\mathcal{O}$ they used can not be constructed under the standard assumption to our best knowledge.

Naturally, there is an open problem: is it possible to construct a (leveled) FHSC scheme from standard assumptions? We give a positive answer in this work.

### A. CONTRIBUTION

Rezaeibagha et al. [3] proposed a new cryptographic concept of the homomorphic signcryption (HSC) and constructed an HSC scheme based on the Decisional Diffie-Hellman (DDH) assumption. However, their scheme is only linearly homomorphic. To overcome linear homomorphism, Li et al. [4] constructed two (leveled) fully homomorphic signcryption (FHSC) schemes. However, their schemes can not be constructed from standard assumption since they employ the indistinguishability obfuscation ($i\mathcal{O}$) which can not be built from standard assumption as far as we know.

In this work, we construct a leveled FHSC from standard lattices by the sign-then-encrypt method, i.e., sign the message first and then encrypt it. In our construction, we utilize the GVW-FHS scheme [2] in the process of signing, while make use of the GSW-FHE scheme [5] in the process of encrypting. Additionally, we prove its completeness, IND-CPA security and strong unforgeability based on the security of GVW and GSW.

It is worth pointing out that the homomorphic multiplication is difficult to work out as we must encrypt each element of the signature matrix, resulting in the FHSC scheme is not practical. It may be possible to improve the efficiency of the scheme by utilizing other FHSes [6], [7], [8], [9] and FHEs [10], [11], [12], [13], however this must overcome the difficulties of homomorphic multiplication.

### B. TECHNIQUE

Given a message $x \in \mathcal{M}$. We initially sign $x$ using GVW to get $\mathbf{U}$. Then we encrypt $\mathbf{U}$ with matrix encryption [14] and encrypt $x$ with GSW scheme. Specifically, we sign it by $\mathbf{V} = \mathbf{AU} + x \cdot \mathbf{G}$, then encrypt $\mathbf{U}$ by $\mathbf{C}_1 = \mathbf{BR} + \begin{pmatrix} \mathbf{U} \\ \mathbf{O} \end{pmatrix}$ and encrypt $x$ by $\mathbf{C}_2 = \mathbf{BF} + x \cdot \mathbf{G}$. Let messages $(x_1, \cdots, x_N)$ and function $f \in \mathcal{F}$, the homomorphic operation needs to satisfy the following conditions:

$$\begin{cases} \mathbf{V}_f = \mathbf{AU}_f + f(x_1, \cdots, x_N) \cdot \mathbf{G} \\ \mathbf{C}_{f,1} = \mathbf{BR}_f + \begin{pmatrix} \mathbf{U_f} \\ \mathbf{O} \end{pmatrix} \\ \mathbf{C}_{f,2} = \mathbf{BF}_f + f(x_1, \cdots, x_N) \cdot \mathbf{G} \end{cases}.$$

Given messages $x_1, x_2 \in \mathcal{M}$. We have that

$$\begin{cases} \mathbf{V}_1 = \mathbf{AU}_1 + x_1 \cdot \mathbf{G} \\ \mathbf{C}_{1,1} = \mathbf{BR}_1 + \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{O} \end{pmatrix} \\ \mathbf{C}_{1,2} = \mathbf{BF}_1 + x_1 \cdot \mathbf{G} \end{cases}$$

and

$$\begin{cases} \mathbf{V}_2 = \mathbf{AU}_2 + x_2 \cdot \mathbf{G} \\ \mathbf{C}_{2,1} = \mathbf{BR}_2 + \begin{pmatrix} \mathbf{U}_2 \\ \mathbf{O} \end{pmatrix} \\ \mathbf{C}_{2,2} = \mathbf{BF}_2 + x_2 \cdot \mathbf{G} \end{cases}.$$

In terms of additive homomorphism, it is easy to get

$$\begin{cases} \mathbf{V}_1 + \mathbf{V}_2 = \mathbf{A}(\mathbf{U}_1 + \mathbf{U}_2) + (x_1 + x_2) \cdot \mathbf{G} \\ \mathbf{C}_{1,1} + \mathbf{C}_{2,1} = \mathbf{B}(\mathbf{R}_1 + \mathbf{R}_2) + \begin{pmatrix} \mathbf{U}_1 + \mathbf{U}_2 \\ \mathbf{O} \end{pmatrix} \\ \mathbf{C}_{1,2} + \mathbf{C}_{2,2} = \mathbf{B}(\mathbf{F}_1 + \mathbf{F}_2) + (x_1 + x_2) \cdot \mathbf{G} \end{cases}.$$

Next we consider multiplicative homomorphism. We have $\mathbf{V}^* = \mathbf{AU}^* + x_1 x_2 \cdot \mathbf{G}$ via utilizing the homomorphic property of GVW, where $\mathbf{U}^* = x_1 \mathbf{U}_2 + \mathbf{U}_2 \mathbf{G}^{-1}(\mathbf{V}_1)$ and $\mathbf{V}^* = \mathbf{V}_2 \mathbf{G}^{-1}(\mathbf{V}_1)$. Let $\mathbf{C}_2^* = \mathbf{C}_{1,2} \cdot \mathbf{G}^{-1}(\mathbf{C}_{2,2})$ and $\mathbf{F}^* = x_1 \mathbf{F}_2 + \mathbf{F}_1 \cdot \mathbf{G}^{-1}(\mathbf{C}_{2,2})$. So, we have $\mathbf{C}_2^* = \mathbf{BF}^* + x_1 x_2 \cdot \mathbf{G}$. Since $\mathbf{U}^* = x_2 \mathbf{U}_1 + \mathbf{U}_2 \mathbf{G}^{-1}(\mathbf{V}_1)$ and we employ two different types of encryption in the process of encrypting, it is difficult for the form of ciphertext to satisfy $\mathbf{C}_1 = \mathbf{BR} + \begin{pmatrix} \mathbf{U} \\ \mathbf{O} \end{pmatrix}$ and $\mathbf{C}_2 = \mathbf{BF} + x \cdot \mathbf{G}$. That is to say, we cannot achieve multiplicative homomorphism operation.

In order to overcome the above issue, we employ the GVW in the process of signing and then apply the technique of encrypting every $u_{i,j}$ of $\mathbf{U}$ to obtain the corresponding encryption matrix by the GSW as it supports simple homomorphic multiplication operations. That is to say, we encrypt it by $\mathbf{C}_{i,j} = \mathbf{BR}_{i,j} + u_{i,j} \cdot \mathbf{G}$ and $\mathbf{C}_2 = \mathbf{BF} + x \cdot \mathbf{G}$. Let messages $(x_1, \cdots, x_N)$ and function $f \in \mathcal{F}$, the homomorphic operation needs to satisfy the following conditions:

$$\begin{cases} \mathbf{V}_f = \mathbf{AU}_f + f(x_1, \cdots, x_N) \cdot \mathbf{G} \\ \mathbf{C}_{f,i,j} = \mathbf{BR}_{f,i,j} + u_{f,i,j} \cdot \mathbf{G} \\ \mathbf{C}_{f,2} = \mathbf{BF}_f + f(x_1, \cdots, x_N) \cdot \mathbf{G} \end{cases},$$

where $\mathbf{C}_{f,1} = \left(\mathbf{C}_{f,i,j}\right)_{i,j \in [m]}$ and $\mathbf{U}_f = \left(u_{f,i,j}\right)_{i,j \in [m]}$.

In the same way, the signatures $\mathbf{U}_1$ and $\mathbf{U}_2$ are encrypted to get the corresponding ciphertext by $\mathbf{C}_{1,i,j} = \mathbf{BR}_{1,i,j} + u_{1,i,j} \cdot \mathbf{G}$ and $\mathbf{C}_{2,i,j} = \mathbf{BR}_{2,i,j} + u_{2,i,j} \cdot \mathbf{G}$, respectively. For the additive operation, we have:

$$\begin{cases} \mathbf{V}_1 + \mathbf{V}_2 = \mathbf{A}(\mathbf{U}_1 + \mathbf{U}_2) + (x_1 + x_2) \cdot \mathbf{G} \\ \mathbf{C}_{1,i,j} + \mathbf{C}_{2,i,j} = \mathbf{B}\left(\mathbf{R}_{1,i,j} + \mathbf{R}_{2,i,j}\right) + \left(u_{1,i,j} + u_{2,i,j}\right)\mathbf{G} \\ \mathbf{C}_{1,2} + \mathbf{C}_{2,2} = \mathbf{B}(\mathbf{F}_1 + \mathbf{F}_2) + (x_1 + x_2) \cdot \mathbf{G} \end{cases}.$$

Following, we consider the homomorphic multiplicative operation. Firstly, we compute homomorphically multiplicative signature to get $\mathbf{U}^* = x_2 \mathbf{U}_1 + \mathbf{U}_2 \mathbf{G}^{-1}(\mathbf{V}_1)$ such that $\mathbf{V}^* = \mathbf{AU}^* + x_1 x_2 \cdot \mathbf{G}$, where $\mathbf{V}^* = \mathbf{V}_2 \mathbf{G}^{-1}(\mathbf{V}_1)$. Secondly, we compute homomorphically the ciphertext of $x_1 x_2$ to obtain

$$\mathbf{C}_2^* = \mathbf{C}_{1,2} \cdot \mathbf{G}^{-1}(\mathbf{C}_{2,2}) = \mathbf{BF}^* + x_1 x_2 \cdot \mathbf{G}$$

where $\mathbf{F}^* = x_1 \mathbf{F}_2 + \mathbf{F}_1 \cdot \mathbf{G}^{-1}(\mathbf{C}_{2,2})$. Lastly, we deal with the encryption of $\mathbf{U}^* = x_2 \mathbf{U}_1 + \mathbf{U}_2 \mathbf{G}^{-1}(\mathbf{V}_1)$ by adding the ciphertext of $x_2 \cdot \mathbf{U}_1$ and the ciphertext of $\mathbf{U}_2 \cdot \mathbf{G}^{-1}(\mathbf{V}_1)$. For

the former, we consider it as a multiplicative homomorphism in the GSW scheme. For the latter, every $\mathbf{C}_{i,j}$ of the ciphertext are obtained by multiplying the corresponding $i$-row of ciphertext of $\mathbf{U}_2$ and $j$-th column of $\mathbf{G}^{-1}(\mathbf{V}_1)$. Then, we have:

$$\mathbf{C}^*_{i,j} = \mathbf{C}_{2,2} \cdot \mathbf{G}^{-1}(\mathbf{C}_{1,1}) + \sum_{k=1}^{m} (\mathbf{C}_{2,1})_{ik} \cdot \mathbf{G}^{-1}(\mathbf{V}_1)_{kj}$$

Let

$$\begin{cases} \mathbf{R}^*_{ij} = \mathbf{F}_2 \cdot \mathbf{G}^{-1}\left(\mathbf{B}\mathbf{R}_{1,i,j} + u_{1,i,j}\right) + x_2\mathbf{R}_{1,i,j} \\ \qquad + \sum_{k=1}^{m} \mathbf{R}_{2,i,k} \cdot \mathbf{G}^{-1}(\mathbf{V}_1)_{kj} \\ u^*_{ij} = x_2 u_{1,i,j} + \sum_{k=1}^{m} u_{2,i,k} \cdot \mathbf{G}^{-1}(\mathbf{V}_1)_{kj} \end{cases},$$

we have

$$\begin{cases} \mathbf{V}^* = \mathbf{A}\mathbf{U}^* + x_1 x_2 \cdot \mathbf{G} \\ \mathbf{C}^*_{ij} = \mathbf{B}\mathbf{R}^*_{ij} + u^*_{ij} \cdot \mathbf{G} \\ \mathbf{C}^*_2 = \mathbf{B}\mathbf{F}^* + x_1 x_2 \cdot \mathbf{G} \end{cases}.$$

Thus, the scheme satisfies homomorphic multiplication. Since $\mathbf{C}^*_{ij} = \mathbf{B}\mathbf{R}^*_{ij} + u^*_{ij} \cdot \mathbf{G}$ and $\mathbf{C}^*_2 = \mathbf{B}\mathbf{F}^* + x_1 x_2 \cdot \mathbf{G}$ which still satisfy the form of GSW. It means that the homomorphic multiplication can continue.

### C. RELATED WORK

#### 1) FULLY HOMOMORPHIC ENCRYPTION

In 2009, Gentry [1] firstly constructed a fully homomorphic encryption (FHE) scheme. Brakerski et al. [10] reduced the magnitude of the noise by the modulus-switching technique in 2012. In the same year, Brakersk [12] present an FHE scheme without modules switching. In 2013, Gentry et al. [5] designed a leveled FHE scheme without computing keys. In their scheme, they employed the "approximate eigenvector" technique. Cheon et al. [13] built a homomorphic encryption for arithmetic of approximate numbers in 2017.

#### 2) FULLY HOMOMORPHIC SIGNATURE

Gorbunov et al. [2] firstly proposed a leveled fully homomorphic signature scheme in 2015. Afterwards, Tsabary [6] testified the equivalence between the homomorphic signature and attribute-based signature. Li et al. [9] built an NTRU-Based FHS scheme. Then Wang et al. [15] designed a leveled strongly-unforgeable identity-based fully homomorphic signature (IBFHS) scheme in 2015. Based on [7] and [15], Wang et al. [16] present leveled IBFHS schemes by using the trapdoor vanishing and vector encoding technique. Wang et al. [8] constructed a more efficient IBFHS scheme whose homomorphic multiplication operation is very similar to the homomorphic addition operation.

#### 3) SIGNCRYPTION

In 1997, Zheng [17] initially introduced a novel cryptographic primitive, called signcryption (SC). In his paper, it could obtain that signcryption can perform both signing and encrypting in a single logical step and its computational as well as communication costs lower than the traditional "sign and encrypt" approach. In 1999, Gamage et.al [18] discussed the public verifiability of signcryption, which is useful for the firewall to verify the validity of the signcryption without decrypting them when they pass through the firewall. Then, Steinfeld et al. [19] proposed a new signcryption scheme based on the integer factorization and testified its unforgeability in the random oracle model in 2000. Baek et.al [20] put forward secure model and constructed a secure signcryption scheme in 2002. In the same year, An et.al [21] also independently gave a secure signcryption model and considered the problem of insider attacks, i.e., the sender can break the privacy of the scheme and the recipient can break the unforgeability of the scheme. And Malone-Lee [22] extended the conception of signcryption to identity-based cryptographic systems in 2002. Then Barbosa and Farshim [23] extended the concept of signcryption to certificateless cryptographic systems in 2008. Subsequently, Liu [24] built secure certificateless signcryption scheme in 2013. In the same year, He et.al [25] constructed a bilinear pair-based signcryption scheme which applied to a remote proof protocol. In 2014, Qi et.al [26] present signcryption scheme with public verifiability and forward security. In the same year, Zhou et al. [27] constructed short signcryption scheme and apply it to Internet of Things. Then Yu et al. [28] proposed a certificateless hybrid signcryption and proved its security in 2015. And then Gao et al. [29] designed a secure certificateless signcryption scheme without bilinear pairing in 2017. Furthermore, there are various schemes have been proposed which combined signcryption with different application purposes or technologies. In recent years, Wang et al. [30] constructed identity-based higncryption scheme, where higncryption means that privacy-enhanced signcryption in 2019. Subsequently, Bellare and Stepanovs [31] stated about security under Message-Derived Keys: Signcryption in iMessage in 2020. Then Liu et al. [32] present Cryptanalysis on 'An efficient identity-based proxy signcryption using lattice' in 2021. And then Hu et.al [33] built a sanitizable signcryption scheme with public verifiability via chameleon hash function in 2022.

### D. ORGANIZATION

The organization of the rest paper is as follows. In Sect.II, we describe some background on lattices and state the related homomorphic schemes. We define the FHSC and related conceptions in Sect.III. In Sect.IV, we construct an FHSC scheme, analyse the variation of noise and prove the security of scheme. Finally, we conclude and proposed some open problems.

## II. PRELIMINARIES

**Basic Notation** $\mathbb{Z}$ denotes the set of integers. Vectors are typed in bold lower-case letters, e.g. $\mathbf{x}$ and matrices are

written in bold capital letters, e.g. $\mathbf{X}$. We utilize the symbol $v \xleftarrow{\$} \mathcal{V}$ to denote the process of opting $v$ from $\mathcal{V}$ uniformly at random. Let $\|\mathbf{A}\|_\infty$ denotes the infinite norm. We use $\lambda$ to denote the security parameter and negl($\lambda$) to denote a negligible function.

### A. BACKGROUND ON THE LATTICES

Let $\mathbf{B} = (b_1, b_2, \cdots, b_n) \subset \mathbb{R}^n$ be $n$ linearly independent vectors. The $n$-dimensional lattice generated by $\mathbf{B}$ is

$$\Lambda = \mathcal{L}(\mathbf{B}) = \{\mathbf{Bc} = \sum_{i=1}^{n} c_i \cdot b_i : c \in \mathbb{Z}^n\}.$$

$\mathbf{B}$ is called as a basis for $\Lambda$.

*Definition 1 ([6] Short Integer Solution Problem):* Let $n$, $m$, $q$ be integer parameters and $\beta > 0$. Given a uniformly matrix $A \in \mathbb{Z}_q^{n\times m}$ at random, find an integer vector $r \in \mathbb{Z}^m$ satisfying $\mathbf{r} \neq 0$ and $\|r\|_\infty \leq \beta$. Then we have $Ar = 0$.

It is widely known that the SIS problem is as difficult as certain worst-case problems in the standard lattices ([34], [35], [36], [37]).

*Definition 2 ([5]$\alpha$-Bounded Distribution):* A distribution ensemble $\{\chi_n\}_{n\in\mathbb{N}}$ supported over the integers, is called $\beta$-bounded if

$$\Pr_{e\leftarrow\chi_n}[|e| > \alpha] = \text{negl}(n).$$

*Definition 3 ([38]Learning With Error Problem):* Let $n$, $m$, $q$ be positive integers and $\chi$ be a distribution over $\mathbb{Z}_q$. Then LWE problem is defined as follows. It is hard to find the vector $\mathbf{s}$ which satisfies $(\mathbf{a}, \langle\mathbf{a}, \mathbf{s}\rangle + \mathbf{e})$ over $\mathbb{Z}_q^n \times \mathbb{Z}_q$, where $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow \chi$ and $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$.

*Lemma 1 ([2], [39], [40], [41], [42]):* There exist several efficient algorithms $\mathsf{TrapGen}$, $\mathsf{SamPre}$, $\mathsf{Sam}$ such that the following holds. Given $n$, $q$ be integers, there exist some $m^*$ and $\beta_{sam}$ so that for all $m \geq m^*$ we have:

1. $\mathbf{U} \leftarrow \mathsf{Sam}(1^m, 1^m, q)$ samples a matrix $\mathbf{U} \in \mathbb{Z}_q^{m\times m}$ which satisfies $\|\mathbf{U}\| \leq \beta_{sam}$.

2. We have the statistical indistinguishability requirements:

$$\mathbf{A} \overset{\text{stat}}{\approx} \mathbf{A}' \quad \textit{and} \quad (\mathbf{A}, \mathbf{T}, \mathbf{U}, \mathbf{V}) \overset{\text{stat}}{\approx} (\mathbf{A}, \mathbf{T}, \mathbf{U}', \mathbf{V}'),$$

where

$$\begin{cases} (\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q) \\ \mathbf{U} \leftarrow \mathsf{Sam}\left(1^m, 1^m, q\right) \\ \mathbf{V} \triangleq \mathbf{A} \cdot \mathbf{U} \end{cases}$$

and

$$\begin{cases} \mathbf{A}' \xleftarrow{\$} \mathbb{Z}_q^{n\times m} \\ \mathbf{U}' \leftarrow \mathsf{SamPre}(\mathbf{A}, \mathbf{V}', \mathbf{T}). \\ \mathbf{V}' \xleftarrow{\$} \mathbb{Z}_q^{n\times m} \end{cases}$$

Therefore, we can obtain that the statistical distance is negligible in $\lambda$.

3. Given n, m, q as above, there is an efficiently and deterministically computable matrix $\mathbf{G} \in \mathbb{Z}_q^{n\times m}$ and a deterministic polynomial-time algorithm $\mathbf{G}^{-1}$ which takes the input $\mathbf{V} \in \mathbb{Z}_q^{n\times m}$ and outputs $\mathbf{R} = \mathbf{G}^{-1}(\mathbf{V})$ such that $\mathbf{R} \in \{0, 1\}^{m\times m}$ and $\mathbf{G} \cdot \mathbf{R} = \mathbf{V}$.

### B. ASSOCIATED HOMOMORPHIC SCHEMES

Here, we give an FHS scheme [2] and an FHE scheme [5], which will be cornerstones of our construction.

**GVW-FHS** [2]. There is an FHS scheme $\mathsf{GVW} = (\mathsf{PrmsGen}, \mathsf{GVW.KeyGen}, \mathsf{GVW.Sign}, \mathsf{SignEval}, \mathsf{GVW.Proc\text{-}ess}, \mathsf{GVW.Verify})$ with massage space $\mathcal{M}$ and output space $\mathcal{V}$,whose algorithms are as follows:

- $\mathsf{GVW.prms} \leftarrow \mathsf{PrmsGen}\left(1^\lambda, 1^N\right)$ : It chooses $(\mathbf{V}_1, \cdots, \mathbf{V}_N)$ by sampling $\mathbf{V}_i \xleftarrow{\$} \mathcal{V}$ and generates parameter $(n_1, m_1, q_1)$. Then it outputs $\mathsf{GVW.prms} = (\mathbf{V}_1, \cdots, \mathbf{V}_N, n_1, m_1, q_1)$.
- $(pk, sk) \leftarrow \mathsf{GVW.KeyGen}(\mathsf{prms})$ : It generates $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{TrapGen}(1^{n_1}, 1^{m_1}, q)$ and sets $pk = \mathbf{A}$, $sk = \mathbf{T}$.
- $(\mathbf{U}_1, \cdots, \mathbf{U}_N) \leftarrow \mathsf{GVW.Sign}_{sk}(x_1, \cdots, x_N)$ : Given $x_i \in \mathcal{M}$ and private key $sk$, it runs $\mathbf{U}_i \leftarrow \mathsf{SamPre}(\mathbf{A}, \mathbf{V}_i - x_i \cdot \mathbf{G}, \mathbf{T})$ satisfying $\mathbf{V}_i = \mathbf{AU}_i + x_i \cdot \mathbf{G}$, where $i \in [N]$.
- $\mathbf{U}^* \leftarrow \mathsf{SignEval}_{pk}(f, (x_1, \mathbf{V}_1, \mathbf{U}_1), \cdots, (x_N, \mathbf{V}_N, \mathbf{U}_N))$ : It inputs $(f, (x_1, \mathbf{V}_1, \mathbf{U}_1), \cdots, (x_N, \mathbf{V}_N, \mathbf{U}_N))$ and outputs $\mathbf{U}^*$.
- $\mathbf{V}_f \leftarrow \mathsf{GVW.Process}_{\mathsf{prms}}(f)$ : It inputs f and $(\mathbf{V}_1, \cdots, \mathbf{V}_N)$ and outputs $\mathbf{V}_f$.
- $0/1 \leftarrow \mathsf{GVW.Verify}_{pk}(\mathbf{V}_f, y, \mathbf{U}^*)$ : If $f_{(pk,y)}(\mathbf{U}^*) = \mathbf{V}_f$, where $y = (x_1, \cdots, x_N)$, then it outputs 1. Otherwise it outputs 0.

*Lemma 2:* Assume that SIS problem is hard, then the GVW-FHS scheme [2] is existentially unforgeable. Furthermore, combining with the strong unforgeability of the Identity-based FHS scheme [15], the GVW-FHS scheme is strongly-unforgeable.

**GSW-FHE** [5]. There is an FHE scheme $\mathsf{GSW} = (\mathsf{GSW.Setup}, \mathsf{GSW.KeyGen}, \mathsf{GSW.Enc}, \mathsf{GSW.Dec}, \mathsf{GSW.Ev\text{-}aluate})$, whose algorithms are as follows:

- $\mathsf{GSW.prms} \leftarrow \mathsf{GSW.Setup}(1^\lambda, 1^L)$ : Given security parameter $\lambda$ and maximum homomorphic depth L, it chooses public parameter $\mathsf{GSW.prms} = (n_2, m_2, q_2, \chi)$.
- $(pk, sk) \leftarrow \mathsf{GSW.KeyGen}(\mathsf{prms})$ : It samples $\mathbf{t} \leftarrow \mathbb{Z}_q^{n_2-1}$ and sets $sk = \mathbf{s} = (1, -\mathbf{t}) \in \mathbb{Z}_q^{n_2}$. Then, It generates a matrix $\mathbf{D} \xleftarrow{\$} \mathbb{Z}_q^{(n_2-1)\times m_2}$ and samples a vector $\mathbf{e} \leftarrow \chi^{m_2}$. It computes $\mathbf{b} = \mathbf{tD} + \mathbf{e}$. Finally, sets $pk = \mathbf{B} = (\mathbf{b}, \mathbf{D}) \in \mathbb{Z}_q^{n_2\times m_2}$. (*Remark* : *Obseve that* $\mathbf{s} \cdot \mathbf{B} = \mathbf{e}$.)
- $\mathbf{C} \leftarrow \mathsf{GSW.Enc}(\mathsf{prms}, \mu, pk)$ : Given a plaintext message $x \in \{0, 1\}$, it samples a uniform matrix $\mathbf{R} \xleftarrow{\$} \{0, 1\}^{m_2\times m_2}$. Computes $\mathbf{C} = \mathbf{B} \cdot \mathbf{R} + x \cdot \mathbf{G} \in \mathbb{Z}_q^{n_2\times m_2}$.
- $0/1 \leftarrow \mathsf{GSW.Dec}(\mathsf{prms}, sk, \mathbf{C})$: Firstly, computes

$$\begin{aligned} \mathbf{s} \cdot \mathbf{C} &= \mathbf{s} \cdot (\mathbf{BR} + x\mathbf{G}) \\ &= \mathbf{e} \cdot \mathbf{R} + x \cdot \mathbf{sG} \end{aligned}$$

Let $\mathbf{c}$ be the first $\lfloor \log q \rfloor$ column of $\mathbf{C}$, then outputs $x = \langle \mathbf{s}, \mathbf{c} \rangle$. Then if the value closes to 0, then outputs 0. Otherwise, outputs 1.

• GSW.Evaluate.

– MultConst($\mathbf{C}, \alpha$). Given a ciphertext $\mathbf{C} \in \mathbb{Z}_q^{n_2 \times m_2}$ and constant $\alpha$, we have

$$\alpha \cdot \mathbf{C} = \mathbf{B} \cdot (\alpha \mathbf{R}) + (\alpha x) \cdot \mathbf{G}.$$

– Add($\mathbf{C}_1, \mathbf{C}_1$). Given two ciphertexts $\mathbf{C}_1, \mathbf{C}_2 \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{R}_1, \mathbf{R}_2 \xleftarrow{\$} \{0,1\}^{m_2 \times m_2}$, we have $\mathbf{C}_1 = \mathbf{B} \cdot \mathbf{R}_1 + x_1 \cdot \mathbf{G}$ and $\mathbf{C}_2 = \mathbf{B} \cdot \mathbf{R}_2 + x_2 \cdot \mathbf{G}$. Thus,

$$\mathbf{C}_1 + \mathbf{C}_2 = \mathbf{B} \cdot (\mathbf{R}_1 + \mathbf{R}_2) + (x_1 + x_2) \cdot \mathbf{G}.$$

– Mult($\mathbf{C}_1, \mathbf{C}_1$). Given two ciphertexts $\mathbf{C}_1, \mathbf{C}_2 \in \mathbb{Z}_q^{n_2 \times m_2}$ and $\mathbf{R}_1, \mathbf{R}_2 \xleftarrow{\$} \{0,1\}^{m_2 \times m_2}$, we have $\mathbf{C}_1 = \mathbf{B} \cdot \mathbf{R}_1 + x_1 \cdot \mathbf{G}$ and $\mathbf{C}_2 = \mathbf{B} \cdot \mathbf{R}_2 + x_2 \cdot \mathbf{G}$. Let $\mathbf{C}^* = \mathbf{C}_1 \mathbf{G}^{-1}(\mathbf{C}_2)$ and $\mathbf{R}^* = \mathbf{R}_1 \mathbf{G}^{-1}(\mathbf{C}_2) + x_1 \mathbf{R}_2$. Thus,

$$\mathbf{C}^* = \mathbf{B} \cdot \mathbf{R}^* + (x_1 \cdot x_2) \cdot \mathbf{G}.$$

*Lemma 3:* Assume that the LWE problem is difficult, then the GSW-FHE scheme [5] is IND-CPA secure.

## III. THE DEFINITION OF FHSC

This section introduces the definition of fully homomorphic signcryption and related conceptions, such as completeness, IND-CPA security and strong unforgeability.

*Definition 4:* A fully homomorphic signcryption scheme is a tuple of algorithms (Setup, KeyGen$_s$, KeyGen$_r$, Signcrypt, Unsigncrypt, Evaluate, Process, Verify) as follows:

• prms $\leftarrow$ Setup$(1^\lambda, 1^L, 1^N)$. Inputs the security parameter $\lambda$, maximum homomorphic depth $L$ and a data-size bound $N$. It generates public parameter prms and defines a message space $\mathcal{M}$.

• $(pk_s, sk_s) \leftarrow$ KeyGen$_s$(prms). It inputs the public parameter prms and outputs a pair of sender's keys $(pk_s, sk_s)$.

• $(pk_r, sk_r) \leftarrow$ KeyGen$_r$(prms). It inputs the public parameter prms and outputs a pair of receiver's keys $(pk_r, sk_r)$.

• $c \leftarrow$ Signcrypt(prms, $sk_s, pk_r, x, i$). It inputs the public parameter prms, sender's private key $sk_s$, receiver's public key $pk_r$, a message $x \in \mathcal{M}$ and its corresponding index $i \in [N]$. Then it outputs a signcryption $c$.

• $x \leftarrow$ Unsigncrypt(prms, $pk_s, sk_r, c$). It inputs the public parameter prms, sender's public key $pk_s$, receiver's private key $sk_r$ and the signcryption $c$. Then it outputs a message $x \in \mathcal{M}$.

• $c^* \leftarrow$ Eval(prms, $pk_s, pk_r, f, \vec{c}$). It inputs the public parameter prms, sender's public key $pk_s$, receiver's public key $pk_r$, a function $f$ and $\vec{c} \in \mathcal{C}^N$. Then it outputs a signcryption $c^*$.

• $v_f \leftarrow$ Process(prms, $f$). It intputs the function $f$ and public parameter prms and outputs $v_f$.

• $0/1 \leftarrow$ Verify(prms, $pk_s, sk_r, x^*, c^*, f$). It inputs the public parameter prms, sender's public key $pk_s$, receiver's public key $pk_r$, receiver's private key $sk_r$, a message $x^* \in \mathcal{M}$, a function $f$ and a signcryption $c^*$. Then it outputs 0 or 1.

*Definition 5 (Completeness):* For prms $\leftarrow$ Setup $(1^\lambda, 1^L, 1^N)$, $(pk_s, sk_s) \leftarrow$ KeyGen$_s$ (prms) and $(pk_r, sk_r) \leftarrow$ KeyGen$_r$(prms), we have:

1. For the $x \in \mathcal{M}$, if $c \leftarrow$ Signcrypt(prms, $sk_s, pk_r, x, i$), then with overwhelming probability it holds that

$$\text{Unsigncrypt}(\text{prms}, pk_s, sk_r, c) = x$$

and

$$\text{Verify}(\text{prms}, pk_s, sk_r, x, c) = 1.$$

2. For $(x_1, \cdots, x_N) \in \mathcal{M}^N$ and a function $f$, if $c_i \leftarrow$ Signcrypt(prms, $sk_s, pk_r, x, i$), where $i \in [N]$ and $c^* \leftarrow$ Eval(prms, $pk_s, pk_r, f, (c_1, \cdots, c_N)$), then with overwhelming probability it holds that

$$\text{Unsigncrypt}(\text{prms}, pk_s, sk_r, c^*) = f(x_1, \cdots, x_N)$$

and

$$\text{Verify}(\text{prms}, pk_s, sk_r, (x_1, \cdots, x_N), c^*, f) = 1.$$

*Definition 6 (IND-CPA Security):* An FHSC scheme is indistinguishable under chosen plaintext attack ($IND-CPA$) if no probabilistic polynomial time (PPT) adversary $\mathcal{A}$ wins the following game with non-negligible advantage.

1. The challenger $\mathcal{C}$ runs Setup to obtain public parameters prms and runs KeyGen$_r$ to obtain receiver's $(pk_r, sk_r)$. Then it gives (prms, $pk_r$) to the adversary $\mathcal{A}$.

2. The adversary $\mathcal{A}$ generates two plaintexts $x_0, x_1 \in \mathcal{M}$ satisfying $|x_0| = |x_1|$ and runs KeyGen$_s$ to obtain sender's $(pk_s, sk_s)$. Then the adversary $\mathcal{A}$ sends $(x_0, x_1, sk_s, pk_s)$ to $\mathcal{C}$. Subsequently, $\mathcal{C}$ chooses a random bit $b \leftarrow \{0,1\}$, runs $c_b \leftarrow$ Signcrypt$(x_b, sk_s, pk_r)$ and sends $c_b$ to $\mathcal{A}$.

3. At the end of the game, $\mathcal{A}$ outputs a bit $b' \leftarrow \{0,1\}$ to $\mathcal{C}$ and wins the game if $b' = b$.

The adversary $\mathcal{A}'s$ advantage in the above game is defined as

$$Adv_{\mathcal{A}}^{IND-CPA} = |Pr[b' = b] - 1/2|.$$

*Definition 7 (Strong Unforgeability):* An FHSC scheme is strongly-unforgeable against chosen message attack if no probabilistic polynomial time (PPT) forger $\mathcal{F}$ wins the following game $\mathbf{Exp}_{\mathcal{F},\text{FHSC}}^{\text{SU-CMA}}$ with non-negligible advantage.

1. The challenger $\mathcal{C}$ runs Setup to obtain public parameters prms. Then the challenger $\mathcal{C}$ runs KeyGen$_s$ to obtain $(pk_s, sk_s)$ and KeyGen$_r$ to obtain $(pk_r, sk_r)$. He sends (prms, $pk_s, pk_r$) to the forger $\mathcal{F}$.

2. The forger $\mathcal{F}$ chooses $(x_1, \cdots, x_N) \in \mathcal{M}$ and gives it to the challenger $\mathcal{C}$. The challenger $\mathcal{C}$ runs $c_i \leftarrow$ Signcrypt(prms, $sk_s, pk_r, x_i$), where $i \in [N]$. And then he sends $(c_1, \cdots, c_N)$ to the forger $\mathcal{F}$.

3. The forger $\mathcal{F}$ chooses a function $f \in \mathcal{F}$ and values $(y', c')$. Then $\mathcal{F}$ sends $(f, y', c')$ to $\mathcal{C}$.

4. The forger $\mathcal{F}$ wins if the following conditions hold.
   – $f$ is admissible on the messages $(x_1, x_2, \cdots, x_N)$.
   – $u' \neq u_f$, where

$$\begin{cases} u' \leftarrow \mathsf{unsigncrypt}(\mathsf{prms}, sk_r, c') \\ c_f \leftarrow \mathsf{Eval}(f, (x_1, c_1), \cdots, (x_N, c_N)) \\ u_f \leftarrow \mathsf{unsigncrypt}(\mathsf{prms}, sk_r, c_f) \end{cases}.$$

   – $\mathsf{Verify}_{pk_s,}(v_f, y', u')$, where $v_f = \mathsf{Process}_{\mathsf{prms}}(f)$.

We say an FHSC is strongly-unforgeable chosen message attack if $\Pr\left[\mathbf{Exp}^{\mathrm{SU-CMA}}_{\mathcal{F},\mathrm{FHSC}}(1^\lambda)\right] < \mathrm{negl}(\lambda)$.
Remark that: Let $y = f(x_1, \cdots, x_N)$. If $y \neq y'$, then $c'$ is a existentially-forgeable signcryption. Otherwise, $c'$ is a strongly-forgeable signcryption.

## IV. THE PROPOSED FHSC SCHEME

In this section, we represent the construction, homomorphic evaluation and security of our fully homomorphic signcryption scheme.

### A. CONSTRUCTION

In this subsection, we describe in detail the construction of FHSC scheme.

**Parameters**. We let $\lambda$ be the security parameter and flexible parameter $L = L(\lambda) = \mathrm{poly}(\lambda)$ be the depth of homomorphism and choose parameters:

$$n, m, q, \beta_{SIS}, \beta_{max}, \beta_{init},$$

depending on $(\lambda, L)$ as follows. Firstly, we set $\beta_{max} \triangleq 2^{\omega(\log\lambda)L}$ and $\beta_{SIS} \triangleq 2^{\omega(\log\lambda)}\beta_{max}$. And then we choose an integer $n = \mathrm{poly}(\lambda, L)$ and a prime $q = 2^{\mathrm{poly}(\lambda,L)} > \beta_{SIS}$ as small as possible so that the $\mathsf{SIS}(n, m, q, \beta_{SIS})$ assumption holds for all $m = \mathrm{poly}(\lambda, L)$. Finally, let $m^* = m^*(n, q) \triangleq O(n \log q)$, $\beta_{sam} = O(n\sqrt{\log q})$ be the parameter required by the trapdoor algorithms $\mathsf{TrapGen}$ as in Lemma 1, and set $m = max\{m^*, n\log q + \omega(\log\lambda)\} = \mathrm{poly}(\lambda, L)$ and $\beta_{init} \triangleq \beta_{sam} = \mathrm{poly}(\lambda, L)$.

Our (leveled) FHSC scheme consists of algorithms ($\mathsf{Setup}$, $\mathsf{KeyGen}_s$, $\mathsf{KeyGen}_r$, $\mathsf{Signcrypt}$, $\mathsf{Unsigncrypt}$, $\mathsf{Eval\text{-}uate}$, $\mathsf{Process}$, $\mathsf{Verify}$) defined as follows:

• $\mathsf{prms} \leftarrow \mathsf{Setup}(1^\lambda, 1^L, 1^N)$. Inputs security parameter $\lambda$, maximum homomorphic depth $L$ and data-size bound $N$. Runs $\mathsf{GVW.prms} \leftarrow \mathsf{PrmsGen}\left(1^\lambda, 1^N\right)$, where $\mathsf{GVW.prms} = (\mathbf{V}_1, \mathbf{V}_2, \cdots, \mathbf{V}_N, n_1, m_1, q_1)$ and runs $\mathsf{GSW.prms} \leftarrow \mathsf{GSW.Setup}\left(1^\lambda, 1^L\right)$, where $\mathsf{GSW.prms} = (n_2, m_2, q_2, \chi)$. Let $n = \max\{n_1, n_2\}$, $m = \max\{m_1, m_2\}$ and $q = \max\{q_1, q_2\}$. Defines domains $\mathcal{M} = \mathbb{Z}_q$, $\mathcal{V} = \mathbb{Z}_q^{n\times m}$. Let $\mathcal{U} = \left\{\mathbf{U} = \mathbb{Z}_q^{m\times m}, \|\mathbf{U}\|_\infty \leqslant \beta_{max}\right\}$. And defines the distribution $\mathbf{U} \leftarrow D_{\mathcal{U}}$ to sample $\mathbf{U} \leftarrow \mathsf{Sam}(1^m, 1^m, q)$ as in Lemma 1 which satisfies $\|\mathbf{U}\|_\infty \leqslant \beta_{init}$. Then outputs $\mathsf{prms} = (\mathbf{V}_1, \mathbf{V}_2, \cdots, \mathbf{V}_N, n, m, q, \chi)$.

• $(pk_s, sk_s) \leftarrow \mathsf{KeyGen}_s(\mathsf{prms})$. Runs $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{GVW.KeyGen}(\mathsf{prms})$ and sets $pk_s = \mathbf{A} \in \mathbb{Z}^{n\times m}$, $sk_s = \mathbf{T}$.

• $(pk_r, sk_r) \leftarrow \mathsf{KeyGen}_r(\mathsf{prms})$. Runs $(\mathbf{B}, \mathbf{s}) \leftarrow \mathsf{GSW.KeyGen}(\mathsf{prms})$ and lets $pk_r = \mathbf{B}$, $sk_r = \mathbf{s}$. Note that $\mathbf{s} = (1, -t) \in \mathbb{Z}_q^m$ and $\mathbf{B} = (\mathbf{b}, \mathbf{D}) \in \mathbb{Z}^{n\times m}$. (Remark that $\mathbf{sB} = \mathbf{e}$.)

• $\mathbf{C} \leftarrow \mathsf{Signcrypt}(\mathsf{prms}, x, sk_s, pk_r)$.
-1. For a message $x \in \mathcal{M}$, runs $\mathbf{U} \leftarrow \mathsf{GVW.Sign}(x, sk_s)$. Remark that $\mathbf{V} = \mathbf{AU} + x \cdot \mathbf{G}$,
where $\mathbf{U} = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1m} \\ u_{21} & u_{22} & \cdots & u_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ u_{m1} & u_{m2} & \cdots & u_{mm} \end{pmatrix}$.
-2. For $\forall u_{ij}$, runs $\mathbf{C}_{ij} \leftarrow \mathsf{GSW.Enc}(\mathsf{prms}, u_{ij}, pk_r)$. Note that $\mathbf{C}_{ij} = \mathbf{B} \cdot \mathbf{R}_{ij} + u_{ij}\mathbf{G}$ $(i, j \in [m])$, where $\mathbf{R}_{ij} \xleftarrow{\$} \{0,1\}^{m\times m}$ $(i, j \in [m])$.
-3. For the message $x \in \mathcal{M}$, runs $\mathbf{C}_2 \leftarrow \mathsf{GSW.Enc}(\mathsf{prms}, x, pk_r)$. Remark that $\mathbf{C}_2 = \mathbf{BF} + x \cdot \mathbf{G}$, where $\mathbf{F} \xleftarrow{\$} \{0,1\}^{m\times m}$.
Then outputs $\mathbf{C} = (\mathbf{C}_1, \mathbf{C}_2)$,
where $\mathbf{C}_1 = \begin{pmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} & \cdots & \mathbf{C}_{1m} \\ \mathbf{C}_{21} & \mathbf{C}_{22} & \cdots & \mathbf{C}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{m1} & \mathbf{C}_{m2} & \cdots & \mathbf{C}_{mm} \end{pmatrix}$.

• $x \leftarrow \mathsf{Unsigncrypt}(\mathsf{prms}, \mathbf{C}, pk_s, sk_r)$. Inputs public parameter $\mathsf{prms}$ and signcryption $\mathbf{C}$, runs algorithm $x \leftarrow \mathsf{GSW.Dec}(\mathbf{C}_2, sk_r)$. Note that

$$\begin{aligned} \mathbf{sC}_2 &= \mathbf{sBF} + x \cdot \mathbf{sG} \\ &= \mathbf{eF} + x \cdot \mathbf{sG}. \end{aligned}$$

Then rounds and outputs the message $x$.

Note that we also can employ algorithm $u_{ij} \leftarrow \mathsf{GSW.Dec}(\mathbf{C}_{ij}, sk_r)$, i.e.,

$$\begin{aligned} \mathbf{sC}_{ij} &= \mathbf{sBR} + u_{ij} \cdot \mathbf{sG} \\ &= \mathbf{eR} + u_{ij} \cdot \mathbf{sG}. \end{aligned}$$

• $\mathbf{C}^* \leftarrow \mathsf{Eval}(\mathsf{prms}, pk_s, pk_r, f, \mathbf{C}_1, \cdots, \mathbf{C}_N)$. Inputs public paramater $\mathsf{prms}$, sender's public key $pk_s$, receiver's public key $pk_r$, function $f$ and $(\mathbf{C}_1, \cdots, \mathbf{C}_N)$ and outputs homomorphic signcryption $\mathbf{C}^*$.

Here, we briefly describe additive homomorphism and multiplicative homomorphism. In the next subsection, we will analyse these characters detailedly.

For $\mathbf{C}_1 = \left(\mathbf{C}_{1,1}, \mathbf{C}_{1,2}\right)$ and $\mathbf{C}_2 = \left(\mathbf{C}_{2,1}, \mathbf{C}_{2,2}\right)$, where $\mathbf{C}_{1,1} = \left(\mathbf{C}_{1,i,j}\right)_{i,j\in[m]}$ and $\mathbf{C}_{2,1} = \left(\mathbf{C}_{2,i,j}\right)_{i,j\in[m]}$.

**Additive Homomorphism**. We define that

$$\begin{aligned} \mathbf{C}_{Add} &= \mathbf{C}_1 + \mathbf{C}_2 \\ &= \left(\mathbf{C}_{1,1} + \mathbf{C}_{2,1}, \mathbf{C}_{1,2} + \mathbf{C}_{2,2}\right). \end{aligned} \tag{1}$$

**Multiplicative Homomorphism**. We define that

$$\begin{aligned} &\mathbf{C}_{Mult} \\ &= \mathbf{C}_1 \odot \mathbf{C}_2 \\ &= \left(\left(\mathbf{C}_{2,2}\mathbf{G}^{-1}(\mathbf{C}_{1,i,j}) + \sum_{k=1}^{m}\left(\mathbf{C}_{2,1}\right)_{ik}\mathbf{G}^{-1}\left(\mathbf{V}_1\right)_{kj}\right)_{i,j\in[m]}, \right. \\ &\quad \left.\mathbf{C}_{1,2}\mathbf{G}^{-1}\left(\mathbf{C}_{2,2}\right)\right). \end{aligned} \tag{2}$$

• $\mathbf{V}_f \leftarrow \mathsf{Process}(\mathsf{prms}, f)$. Inputs function $f$ and public parameter $\mathsf{prms}$. Then runs $\mathbf{V}_f \leftarrow \mathsf{GVW.Process}_{\mathsf{prms}}(f)$ and outputs $\mathbf{V}_f$.

We define additive homomorphism and multiplicative homomorphism in the $\mathsf{Process}$ algorithm as follows.

**Addition**. For $\mathbf{V}_1$ and $\mathbf{V}_2$, we have $\mathbf{V}_{Add} = \mathbf{V}_1 + \mathbf{V}_2$.

**Multiplication**. For $\mathbf{V}_1$ and $\mathbf{V}_2$, we have $\mathbf{V}_{Mult} = \mathbf{V}_2 \cdot \mathbf{G}^{-1}(\mathbf{V}_1)$.

• $0/1 \leftarrow \mathsf{Verify}(\mathsf{prms}, pk_s, sk_r, \mathbf{C}^*, f)$. Taking as input public parameter $\mathsf{prms}$, sender's public key $pk_s$, receiver's private key $sk_r$, the signcryption $\mathbf{C}^* = (\mathbf{C}_1^*, \mathbf{C}_2^*)$ where $\mathbf{C}_1^* = \left(\mathbf{C}_{ij}^*\right)_{i,j\in[m]}$ and a function $f \in \mathcal{F}$. Runs algorithms $x^* \leftarrow \mathsf{GSW.Dec}(\mathsf{prms}, \mathbf{C}_2^*, sk_r)$ and $u_{ij}^* \leftarrow \mathsf{GSW.Dec}(\mathsf{prms}, \mathbf{C}_{ij}^*, sk_r)$. Note that

$$\begin{aligned}\mathbf{sC}_2^* &= \mathbf{sBF}^* + x^* \cdot \mathbf{sG}\\ &= \mathbf{eF}^* + x^* \cdot \mathbf{sG}\end{aligned}$$

and

$$\begin{aligned}\mathbf{sC}_{ij}^* &= \mathbf{sBR}_{ij}^* + u_{ij}^* \cdot \mathbf{sG}\\ &= \mathbf{eR}_{ij}^* + u_{ij}^* \cdot \mathbf{sG},\end{aligned}$$

where $\left(\mathbf{C}_{ij}^*\right)_{i,j\in[m]} = \mathbf{C}_1^*$ and $\left(u_{ij}^*\right)_{i,j\in[m]} = \mathbf{U}^*$. Then rounds and outputs the signature $\mathbf{U}^*$ and the message $x^*$. Lastly, runs $0/1 \leftarrow \mathsf{GVW.Verify}(\mathsf{prms}, pk_s, x^*, \mathbf{U}^*)$.

Remark that: if $\mathbf{V}^* = \mathbf{A} \cdot \mathbf{U}^* + x^* \cdot \mathbf{G}$, then outputs 1. Otherwise outputs 0.

### B. HOMOMORPHIC EVALUATION

In this subsection, we demonstrate that the signcryption form (including signature form) is kept after the homomorphic addition and multiplication. This helps us to analyse the variation of noise-level before and after the homomorphic addition and multiplication.

For signcryptions $\mathbf{C}_1 = (\mathbf{C}_{1,1}, \mathbf{C}_{1,2})$ and $\mathbf{C}_2 = (\mathbf{C}_{2,1}, \mathbf{C}_{2,2})$, where $\mathbf{C}_{1,1} = (\mathbf{C}_{1,i,j})_{i,j\in[m]}$, $\mathbf{C}_{2,1} = (\mathbf{C}_{2,i,j})_{i,j\in[m]}$ and

$$\begin{cases}\mathbf{C}_{1,i,j} = \mathbf{BR}_{1,i,j} + u_{1,i,j} \cdot \mathbf{G}\ (i,j \in [m])\\ \mathbf{C}_{2,i,j} = \mathbf{BR}_{2,i,j} + u_{2,i,j} \cdot \mathbf{G}\ (i,j \in [m])\\ \mathbf{C}_{1,2} = \mathbf{BF}_1 + x_1 \cdot \mathbf{G}\\ \mathbf{C}_{2,2} = \mathbf{BF}_2 + x_2 \cdot \mathbf{G}.\end{cases} \tag{3}$$

For the signatures $\mathbf{U}_1 = (u_{1,i,j})_{i,j\in[m]}$ and $\mathbf{U}_2 = (u_{2,i,j})_{i,j\in[m]}$, it implies that $\mathbf{V}_1 = \mathbf{AU}_1 + x_1 \cdot \mathbf{G}$ and $\mathbf{V}_2 = \mathbf{AU}_2 + x_2 \cdot \mathbf{G}$.

**Homomorphic Addition**. Addition to (1), we can compute that

$$\begin{aligned}\mathbf{C}_{Add} &= \mathbf{C}_1 + \mathbf{C}_2\\ &= (\mathbf{C}_{1,1} + \mathbf{C}_{2,1}, \mathbf{C}_{1,2} + \mathbf{C}_{2,2})\\ &= \left((\mathbf{B}(\mathbf{R}_{1,i,j} + \mathbf{R}_{2,i,j}) + (u_{1,i,j} + u_{2,i,j}) \cdot \mathbf{G})_{i,j\in[m]},\right.\\ &\quad \left.\mathbf{B}(\mathbf{F}_1 + \mathbf{F}_2) + (x_1 + x_2) \cdot \mathbf{G}\right).\end{aligned}$$

Remark that, addition to $\mathsf{Process}(\mathsf{prms}, f)$, it implies that

$$\begin{aligned}\mathbf{V}_{Add} &= \mathbf{V}_1 + \mathbf{V}_2\\ &= (\mathbf{AU}_1 + x_1 \cdot \mathbf{G}) + (\mathbf{AU}_2 + x_2 \cdot \mathbf{G})\\ &= \mathbf{A}(\mathbf{U}_1 + \mathbf{U}_2) + (x_1 + x_2) \cdot \mathbf{G}\\ &= \mathbf{AU}_{Add} + (x_1 + x_2) \cdot \mathbf{G}.\end{aligned}$$

Define $\mathbf{C}_{Add,1} = \left(\mathbf{C}_{Add1,i,j}\right)_{i,j\in[m]} = (\mathbf{B}(\mathbf{R}_{1,i,j} + \mathbf{R}_{2,i,j}) + (u_{1,i,j} + u_{2,i,j}) \cdot \mathbf{G})_{i,j\in[m]}$, $\mathbf{R}_{Add1,i,j} = \mathbf{R}_{1,i,j} + \mathbf{R}_{2,i,j}$, $\mathbf{C}_{Add,2} = \mathbf{B}(\mathbf{F}_1 + \mathbf{F}_2) + (x_1 + x_2) \cdot \mathbf{G}$ and $\mathbf{F}_{Add} = \mathbf{F}_1 + \mathbf{F}_2$, we have

$$\begin{cases}\mathbf{V}_{Add} = \mathbf{AU}_{Add} + (x_1 + x_2) \cdot \mathbf{G}\\ \quad \mathbf{C}_{Add1,i,j} = \mathbf{BR}_{Add1,i,j} + (u_{1,i,j} + u_{2,i,j}) \cdot \mathbf{G}(i,j \in [m])\ .\\ \mathbf{C}_{Add,2} = \mathbf{BF}_{Add} + (x_1 + x_2) \cdot \mathbf{G}\end{cases}$$

That is to say, homomorphic addition satisfies the form of signcryption. In the process of unsigncrypting, we compute that

$$\begin{aligned}\mathbf{sC}_{Add1,i,j} &= \mathbf{sBR}_{Add1,i,j} + (u_{1,i,j} + u_{2,i,j}) \cdot \mathbf{sG}\\ &= \mathbf{eR}_{Add1,i,j} + (u_{1,i,j} + u_{2,i,j}) \cdot \mathbf{sG}\end{aligned}$$

and

$$\begin{aligned}\mathbf{sC}_{Add,2} &= \mathbf{sBF}_{Add} + (x_1 + x_2) \cdot \mathbf{sG}\\ &= \mathbf{eF}_{Add} + (x_1 + x_2) \cdot \mathbf{sG}.\end{aligned}$$

Now, we analyse the corresponding noise-level in the process of additive homomorphism. If $\mathbf{U}_1$ and $\mathbf{U}_2$ are bounded by $\beta$, then it is easy to get that $\mathbf{U}_{Add}$ is bounded by $2\beta$. If $\mathbf{eR}_{1,i,j}$, $\mathbf{eR}_{2,i,j}$, $\mathbf{eF}_1$and $\mathbf{eF}_2$ are bounded by $\alpha$, then it is also easy to get that $\mathbf{C}_{Add}$ is bounded by $2\alpha$.

**Homomorphic Multiplication**. Addition to (2), we can compute that

$$\begin{aligned}&\mathbf{C}_{Mult}\\ &= \mathbf{C}_1 \odot \mathbf{C}_2\\ &= (\mathbf{C}_{Mult1,i,j}, \mathbf{C}_{Mult,2})\\ &= \left(\left(\mathbf{C}_{2,2}\mathbf{G}^{-1}(\mathbf{C}_{1,i,j}) + \sum_{k=1}^{m}(\mathbf{C}_{2,1})_{ik}\,\mathbf{G}^{-1}(\mathbf{V}_1)_{kj}\right)_{i,j\in[m]},\right.\\ &\quad \left.\mathbf{C}_{1,2}\mathbf{G}^{-1}(\mathbf{C}_{2,2})\right),\end{aligned}$$

where

$$\begin{aligned}&\mathbf{C}_{Mult1,i,j}\\ &\triangleq \left(\mathbf{C}_{2,2}\mathbf{G}^{-1}(\mathbf{C}_{1,i,j}) + \sum_{k=1}^{m}(\mathbf{C}_{2,1})_{ik}\,\mathbf{G}^{-1}(\mathbf{V}_1)_{kj}\right)_{i,j\in[m]}\\ &= (\mathbf{BF}_2 + x_2\mathbf{G})\,\mathbf{G}^{-1}(\mathbf{C}_{1,i,j})\\ &\quad + \sum_{k=1}^{m}\left(\mathbf{BR}_{2,i,k} + u_{2,i,k}\mathbf{G}\right)\,\mathbf{G}^{-1}(\mathbf{V}_1)_{kj}\\ &= \mathbf{BF}_2\mathbf{G}^{-1}\left(\mathbf{BR}_{1,i,j} + u_{1,i,j}\mathbf{G}\right) + x_2\mathbf{BR}_{1,i,j} + x_2u_{1,i,j}\mathbf{G}\\ &\quad + \sum_{k=1}^{m}\mathbf{BR}_{2,i,k}\cdot\mathbf{G}^{-1}(\mathbf{V}_1)_{kj} + \sum_{k=1}^{m}u_{2,i,k}\mathbf{G}^{-1}(\mathbf{V}_1)_{kj}\mathbf{G}\\ &= \mathbf{B}(\mathbf{F}_2\mathbf{G}^{-1}\left(\mathbf{BR}_{1,i,j} + u_{1,i,j}\right) + x_2\mathbf{R}_{1,i,j}\\ &\quad + \sum_{k=1}^{m}\mathbf{R}_{2,i,k}\mathbf{G}^{-1}(\mathbf{V}_1)_{kj})\end{aligned}$$

$$+\left(x_2u_{1,i,j}+\sum_{k=1}^{m}u_{2,i,k}\cdot\mathbf{G}^{-1}(\mathbf{V}_1)_{kj}\right)\mathbf{G}$$

and

$$\begin{aligned}\mathbf{C}_{Mult,2}&\triangleq\mathbf{C}_{1,2}\mathbf{G}^{-1}\left(\mathbf{C}_{2,2}\right)\\&=\mathbf{B}\left(x_1\mathbf{F}_2+\mathbf{F}_1\cdot\mathbf{G}^{-1}(\mathbf{C}_{2,2})\right)+x_1x_2\mathbf{G}.\end{aligned}$$

Remark that: Let

$$\begin{cases}\mathbf{R}_{Mult1,i,j}\triangleq\mathbf{F}_2\mathbf{G}^{-1}(\mathbf{B}\mathbf{R}_{1,i,j}+u_{1,i,j}\mathbf{G})+x_2\mathbf{R}_{1,i,j}\\\qquad\qquad+\sum_{k=1}^{m}\mathbf{R}_{2,i,k}\mathbf{G}^{-1}(\mathbf{V}_1)_{kj}\\u_{Mult1,i,j}\triangleq x_2u_{1,i,j}+\sum_{k=1}^{m}u_{2,i,k}\mathbf{G}^{-1}(\mathbf{V}_1)_{kj}\\\mathbf{F}_{Mult}\triangleq x_1\mathbf{F}_2+\mathbf{F}_1\cdot\mathbf{G}^{-1}(\mathbf{C}_{2,2})\end{cases},$$

we have

$$\begin{cases}\mathbf{C}_{Mult1,i,j}=\mathbf{B}\mathbf{R}_{Mult1,i,j}+u_{Mult1,i,j}\cdot\mathbf{G}\\\mathbf{C}_{Mult,2}=\mathbf{B}\mathbf{F}_{Mult}+x_1x_2\cdot\mathbf{G}\end{cases}.$$

Remark that, addition to $\mathsf{Process}(\mathsf{prms},f)$, it implies that

$$\begin{aligned}\mathbf{V}_{Mult}&=\mathbf{V}_2\mathbf{G}^{-1}\left(\mathbf{V}_1\right)\\&=(\mathbf{A}\mathbf{U}_2+x_2\mathbf{G})\mathbf{G}^{-1}\left(\mathbf{A}\mathbf{U}_1+x_1\mathbf{G}\right)\\&=\mathbf{A}(x_2\mathbf{U}_1+\mathbf{U}_2\mathbf{G}^{-1}(\mathbf{V}_1))+(x_1x_2)\mathbf{G}\\&=\mathbf{A}\mathbf{U}_{Mult}+(x_1x_2)\mathbf{G}.\end{aligned}$$

Thus, multiplicative homomorphism satisfies the form of signcryption.

*Theorem 1:* If signcryption $\mathbf{C}_1=\left(\mathbf{C}_{1,1},\mathbf{C}_{1,2}\right)$ and $\mathbf{C}_2=\left(\mathbf{C}_{2,1},\mathbf{C}_{2,2}\right)$ defined by (3) are bounded by $\alpha$. Then it holds that

(1). The noise-level of $\mathbf{C}_{Mult}=\mathbf{C}_1\odot\mathbf{C}_2$ is bounded by $(m^2+m+1)\alpha$.

(2). If $\mathbf{U}_1$ and $\mathbf{U}_2$ are bounded by $\beta$, i.e.$\|\mathbf{U}_1\|_\infty,\|\mathbf{U}_2\|_\infty\leq\beta$, then the corresponding signature $\mathbf{U}_{Mult}$ of $\mathbf{C}_{Mult}$ is bounded by $(m+1)\beta$.

*Proof 1:* (1). Since the signcryptions $\mathbf{C}_1=\left(\mathbf{C}_{1,1},\mathbf{C}_{1,2}\right)$ and $\mathbf{C}_2=\left(\mathbf{C}_{2,1},\mathbf{C}_{2,2}\right)$ are bounded by $\alpha$, then $\mathbf{C}_{1,1}$, $\mathbf{C}_{2,1}$, $\mathbf{C}_{1,2}$ and $\mathbf{C}_{2,2}$ are bounded by $\alpha$. Owing to $\mathbf{C}_{1,1}=\left(\mathbf{C}_{1,i,j}\right)_{i,j\in[m]}$ and $\mathbf{C}_{2,1}=\left(\mathbf{C}_{2,i,j}\right)_{i,j\in[m]}$, we can obtain $\mathbf{C}_{1,i,j}$ and $\mathbf{C}_{2,i,j}$ are bounded by $\alpha$.

That is to say, $\|\mathbf{e}\mathbf{R}_{1,i,j}\|_\infty,\|\mathbf{e}\mathbf{R}_{2,i,j}\|_\infty,\|\mathbf{e}\mathbf{F}_1\|_\infty,\|\mathbf{e}\mathbf{F}_2\|_\infty\leq\alpha$. Let the signcryption defined by

$$\mathbf{C}_{Mult}=\mathbf{C}_1\odot\mathbf{C}_2=\left(\mathbf{C}_{Mult,1},\mathbf{C}_{Mult,2}\right).$$

First of all, for the signcryption $\mathbf{C}_{Mult,1}$, we compute this noise-level as follows:

$$\begin{aligned}&\|\mathbf{e}\mathbf{R}_{Mult1,i,j}\|_\infty\\&=\|\mathbf{e}(\mathbf{F}_2\mathbf{G}^{-1}(\mathbf{B}\mathbf{R}_{1,i,j}+u_{1,i,j}\mathbf{G})+x_2\mathbf{R}_{1,i,j}\\&\quad+\sum_{k=1}^{m}\mathbf{R}_{2,i,k}\mathbf{G}^{-1}(\mathbf{V}_1)_{kj})\|_\infty\\&=\|\mathbf{e}\mathbf{F}_2\mathbf{G}^{-1}(\mathbf{B}\mathbf{R}_{1,i,j}+u_{1,i,j}\mathbf{G})+x_2\mathbf{e}\mathbf{R}_{1,i,j}\\&\quad+\sum_{k=1}^{m}\mathbf{e}\mathbf{R}_{2,i,k}\mathbf{G}^{-1}(\mathbf{V}_1)_{kj}\|_\infty\\&\leq\|\mathbf{e}\mathbf{F}_2\mathbf{G}^{-1}(\mathbf{B}\mathbf{R}_{1,i,j}+u_{1,i,j}\mathbf{G})\|_\infty+\|x_2\mathbf{e}\mathbf{R}_{1,i,j}\|_\infty\\&\quad+\|\sum_{k=1}^{m}\mathbf{e}\mathbf{R}_{2,i,k}\mathbf{G}^{-1}(\mathbf{V}_1)_{kj}\|_\infty\\&\leq m\alpha+\alpha+m^2\alpha\\&=(m^2+m+1)\alpha.\end{aligned}$$

Thus, $\|\mathbf{e}\mathbf{R}_{Mult,1}\|_\infty=max\{\|\mathbf{e}\mathbf{R}_{Mult1,i,j}\|_\infty\}\leq(m^2+m+1)\alpha$.

Secondly, for the signcryption $\mathbf{C}_{Mult,2}$, we compute this noise-level as follows:

$$\begin{aligned}\|\mathbf{e}\mathbf{F}_{Mult}\|_\infty&=\|\mathbf{e}(x_1\mathbf{F}_2+\mathbf{F}_1\cdot\mathbf{G}^{-1}(\mathbf{C}_{2,2}))\|_\infty\\&=\|x_1\mathbf{e}\mathbf{F}_2+\mathbf{e}\mathbf{F}_1\cdot\mathbf{G}^{-1}(\mathbf{C}_{2,2})\|_\infty\\&\leq\|x_1\mathbf{e}\mathbf{F}_2\|_\infty+\|\mathbf{e}\mathbf{F}_1\cdot\mathbf{G}^{-1}(\mathbf{C}_{2,2})\|_\infty\\&\leq\alpha+m\alpha\\&=(m+1)\alpha.\end{aligned}$$

Finally, we can get

$$\begin{aligned}\alpha_{Mult}&\leqslant\max\left\{\|\mathbf{e}\mathbf{R}_{Mult,1}\|_\infty,\|\mathbf{e}\mathbf{F}_{Mult}\|_\infty\right\}\\&\leqslant\max\left\{\left(m^2+m+1\right)\alpha,(m+1)\alpha\right\}\\&\leqslant\left(m^2+m+1\right)\alpha.\end{aligned}$$

That is to say, the noise-level of $\mathbf{C}_{Mult}$ will be $\left(m^2+m+1\right)\alpha$.

(2). We have $\mathbf{U}_{Mult}=x_2\mathbf{U}_1+\mathbf{U}_2\mathbf{G}^{-1}(\mathbf{V}_1)$, then the noise-level as follows.

$$\begin{aligned}\|\mathbf{U}_{Mult}\|_\infty&=\|x_2\mathbf{U}_1+\mathbf{U}_2\mathbf{G}^{-1}(\mathbf{V}_1)\|_\infty\\&\leq\|x_2\mathbf{U}_1\|_\infty+\|\mathbf{U}_2\mathbf{G}^{-1}(\mathbf{V}_1)\|_\infty\\&\leq\beta+m\beta\\&=(m+1)\beta.\end{aligned}$$

Next, we discuss about the noise-level on the admissible function $f$ with depth $L$. If $\mathbf{U}_t$ is bounded by $\beta_{init}$, then $\mathbf{U}^*$ will be bounded by $\beta^*\leqslant\beta_{init}\cdot(m+1)^L\leqslant2^{\mathrm{O}(\log\lambda)\cdot L}\leqslant\beta_{max}$. If $\mathbf{e}\mathbf{R}_{t,i,j}$ and $\mathbf{e}\mathbf{F}_t$ are bounded by $\alpha$, then $\mathbf{C}^*$ will be bounded by $\alpha^*\leqslant\left(m^2+m+1\right)^L\alpha$.

**Faster Homomorphic Multiplication**. In addition, it is very easy to see that the noise-level of $\mathbf{C}_{Mult,1}$ is more influential than $\mathbf{C}_{Mult,2}$ in the process of multiplicative homomorphism. So, we can employ this partial property when computing an $l$-degree monomial.

*Theorem 2:* Given fresh signcryptions $\mathbf{C}_1,\mathbf{C}_2,\cdots,\mathbf{C}_l$, where $\mathbf{C}_i=(\mathbf{C}_{i,1},\mathbf{C}_{i,2})(i=1,\cdots,l)$. Then it holds that

(1). The noise-level of $\mathbf{C}^*=\mathbf{C}_1\odot\mathbf{C}_2\odot\cdots\odot\mathbf{C}_l=(\cdots((\mathbf{C}_1\odot\mathbf{C}_2)\odot\mathbf{C}_3)\cdots)\odot\mathbf{C}_l$ is bounded by $(l-1)(m^2+m+1)\alpha$.

(2). The corresponding signature $\mathbf{U}^*$ of $\mathbf{C}^*$ is bounded by $(l-1)(m+1)\beta$.

*Proof 2:* (1). For the noise-level in the two-multiplication, let $\mathbf{e}\mathbf{R}_{Mult1,i,j}$ be bounded by $(m^2+m+1)\alpha$ as in

Theorem 1 and $\mathbf{eF}_3$ and $\mathbf{eR}_{3,i,j}$ be bounded by $\alpha$. Let $\mathbf{C}_{2Mult} = ((\mathbf{C}_{2Mult,1})_{i,j\in[m]}, \mathbf{C}_{2Mult,2})$. Then $\mathbf{eR}_{2Mult,i,j}$ be the noise-level of $(\mathbf{C}_{2Mult,1})_{i,j\in[m]}$ and $\mathbf{eF}_{2Mult}$ be the noise-level of $\mathbf{C}_{2Mult,2}$.

Firstly, we can get the noise-level of $(\mathbf{C}_{2Mult,1})_{i,j\in[m]}$ as follows.

$$\begin{aligned}
&\|\mathbf{eR}_{2Mult,i,j}\|_\infty \\
&= \|\mathbf{e}(\mathbf{F}_3\mathbf{G}^{-1}(\mathbf{BR}_{2,i,j} + u_{2,i,j}\mathbf{G}) + x_3\mathbf{R}_{Mult1,i,j} \\
&\quad + \sum_{k=1}^{m} \mathbf{R}_{3,i,k}\mathbf{G}^{-1}(\mathbf{V}_2)_{kj})\|_\infty \\
&= \|\mathbf{eF}_3\mathbf{G}^{-1}(\mathbf{BR}_{2,i,j} + u_{2,i,j}\mathbf{G}) + x_3\mathbf{eR}_{Mult1,i,j} \\
&\quad + \sum_{k=1}^{m} \mathbf{eR}_{3,i,k}\mathbf{G}^{-1}(\mathbf{V}_2)_{kj}\|_\infty \\
&\le \|\mathbf{eF}_3\mathbf{G}^{-1}(\mathbf{BR}_{2,i,j} + u_{1,i,j}\mathbf{G})\|_\infty + \|x_3\mathbf{eR}_{Mult1,i,j}\|_\infty \\
&\quad + \|\sum_{k=1}^{m} \mathbf{eR}_{3,i,k}\mathbf{G}^{-1}(\mathbf{V}_2)_{kj}\|_\infty \\
&\le m\alpha + (m^2 + m + 1)\alpha + m^2\alpha \\
&= (2m^2 + 2m + 1)\alpha \\
&\le 2(m^2 + m + 1)\alpha.
\end{aligned}$$

Secondly, we can obtain the noise-level of $\mathbf{C}_{2Mult,2}$ as follows.

$$\begin{aligned}
\|\mathbf{eF}_{2Mult}\|_\infty &= \|\mathbf{e}\left(x_{Mult}\mathbf{F}_3 + \mathbf{F}_{Mult}\cdot\mathbf{G}^{-1}(\mathbf{C}_{3,2})\right)\|_\infty \\
&= \|x_{Mult}\mathbf{eF}_3 + \mathbf{eF}_{Mult}\cdot\mathbf{G}^{-1}(\mathbf{C}_{3,2})\|_\infty \\
&\le \|x_{Mult}\mathbf{eF}_3\|_\infty + \|\mathbf{eF}_{Mult}\cdot\mathbf{G}^{-1}(\mathbf{C}_{3,2})\|_\infty \\
&\le \alpha + (m+1)\alpha \\
&= (2m+1)\alpha \\
&\le 2(m+1)\alpha.
\end{aligned}$$

Thus, we can get the noise-level of $\mathbf{C}_{Mult2}$ as follows.

$$\begin{aligned}
\alpha_{2Mult} &\leqslant \max\left\{\|\mathbf{eR}_{2Mult,i,j}\|_\infty, \|\mathbf{eF}_{2Mult}\|_\infty\right\} \\
&\leqslant \max\left\{2(m^2+m+1)\alpha, 2(m+1)\alpha\right\} \\
&\leqslant 2(m^2+m+1)\alpha.
\end{aligned}$$

That is to say, we can get the noise-level of $\mathbf{C}_{(l-1)Mult}$ is equivalent to the noise-level of the former. Addition to recursion, we can obtain the corresponding noise-level of signcryption $\mathbf{C}_{(l-1)Mult} = \mathbf{C}_1 \odot \mathbf{C}_2 \odot \cdots \odot \mathbf{C}_l$ is bounded by $(l-1)(m^2 + m + 1)\alpha$.

(2). Addition to recursion and Lemma 1, we can obtain that the noise-level of $\mathbf{U}^*$ is bounded by $(l-1)(m+1)\beta$.

In this case, we can set smaller parameters accordingly.

**Completeness**. Addition to the noise-level analysed on the above, we could set appropriate parameters in order to guarantee the correctness of homomorphic evaluation, which in turn assure the completeness of the proposed FHSC scheme.

### C. SECURITY

*Theorem 3 (IND-CPA Security):* The proposed FHSC scheme is IND-CPA secure, if the GSW scheme is IND-CPA secure.

*Proof 3:* From the construction of FHSC scheme, our signcryption are exactly $(m^2 + 1)$ ciphertexts of GSW. Thus, the IND-CPA security of FHSC scheme is directly derived from the IND-CPA security of GSW.

*Theorem 4 (Strong Unforgeability):* Our (*leveled*) FHSC scheme is strongly unforgeable, if the GSW scheme is IND-CPA secure and GVW scheme is strongly unforgeable.

*Proof 4:* We assume that exists a PPT forger $\mathcal{F}$ can forge a signcryption with non-negligible probability $\delta$. Then $\mathcal{F}$ can break the IND-CPA security of GSW or the strong-unforgeability of GVW.

The reduction $\mathcal{C}$ runs Setup to obtain public parameter prms and generates a pair of sender's key $(\mathbf{A}, \mathbf{T})$ as well as a pair of receiver's key $(\mathbf{B}, \mathbf{s})$. Then he sends $(\text{prms}, \mathbf{A}, \mathbf{B})$ to $\mathcal{F}$. In the process of signcryption querying, $\mathcal{F}$ chooses messages $(x_1, \cdots, x_N)$ and sends them to $\mathcal{C}$. Then $\mathcal{C}$ signcrypts messages to obtain the signcryptions $(\mathbf{C}_1, \cdots, \mathbf{C}_N)$ and sends them to $\mathcal{F}$.

Assume that the forger $\mathcal{F}$ can forge a signcryption. That is to say, the forger $\mathcal{F}$ sends a tuple of forgery $(f, y', \mathbf{C}')$ to $\mathcal{C}$, where $\mathbf{C}' = (\mathbf{C}'_1, \mathbf{C}'_2)$, $\mathbf{C}'_1 = \left(\mathbf{C}'_{ij}\right)_{i,j\in[m]}$.

Firstly, f is an admissible function. If not, $(f, y', \mathbf{C}')$ is not a forgery. Secondly, $\mathcal{C}$ consider forger's signcryption $\mathbf{C}'$. $\mathcal{C}$ unsigncrypts the forger's signcryption $\mathbf{C}'$ to obtain the corresponding signature $\mathbf{U}'$. Then the $\mathcal{C}$ computes $\mathbf{C}_f = \text{Eval}_{\text{prms}}(f, (x_1, \mathbf{C}_1), \cdots, (x_N, \mathbf{C}_N))$ and unsigncrypts the signcryption $\mathbf{C}_f = (\mathbf{C}_{f,1}, \mathbf{C}_{f,2})$ to obtain the corresponding signature $\mathbf{U}_f$. If $\mathbf{U}' = \mathbf{U}_f$, it means that the forger can decrypt the homomorphic signcryption $\mathbf{C}_f$. That is to say, $\mathcal{F}$ breaks the IND-CPA security of GSW.

Otherwise, we have $\mathbf{U}_f \neq \mathbf{U}'$. Let $y = f(x_1, \cdots, x_N)$ and $\mathbf{V}_f = \text{Process}_{\text{prms}}(f)$. It holds that $\mathbf{V}_f = \mathbf{AU}' + y' \cdot G = \mathbf{AU}_f + y \cdot G$. Thus, $\mathbf{U}'$ is indeed a forgery of GVW. In other words, $\mathcal{F}$ breaks the strong unforgeability of GVW.

## V. CONCLUSION AND OPEN PROBLEMS

In this paper, we constructed a leveled FHSC scheme and proved its completeness, IND-CPA security and strong-unforgeability under the standard assumption. However, the proposed FHSC scheme is not efficient enough. Therefore, it is interesting to design more efficient FHSC schemes. Moreover, it is an important problem to construct non-leveled FHSC schemes. It is well known that there have been non-leveled FHE schemes using bootstrapping. Thus, it maybe a method to design non-leveled FHS schemes firstly, and then transport it to non-leveled FHSC schemes.

## REFERENCES

[1] C. Gentry. (2009). *A Fully Homomorphic Encryption Scheme (Ph.D. Thesis)*. [Online]. Available: http://crypto.stanford.edu/craig/

[2] S. Gorbunov, V. Vaikuntanathan, and D. Wichs, "Leveled fully homomorphic signatures from standard lattices," in *Proc. 47th Annu. ACM Symp. Theory Comput.*, Jun. 2015, pp. 469–477, doi: 10.1145/2746539.2746576.

[3] F. Rezaeibagha, Y. Mu, S. Zhang, and X. Wang, "Provably secure homomorphic signcryption," in *Proc. Int. Conf. Provable Secur.*, in Lecture Notes in Computer Science, vol. 10592, 2017, pp. 349–360, doi: 10.1007/978-3-319-68637-0_21.

[4] S. Li, B. Liang, A. Mitrokotsa, and R. Xue, "Homomorphic signcryption with public plaintext-result checkability," *IET Inf. Secur.*, vol. 15, no. 5, pp. 333–350, Sep. 2021, doi: 10.1049/ise2.12026.

[5] C. Gentry, A. T. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," in *Advances in Cryptology—CRYPTO 2013*, vol. 8042. Heidelberg, Germany: Springer, 2013, pp. 75–92.

[6] R. Tsabary, "An equivalence between attribute-based signatures and homomorphic signatures, and new constructions for both," in *Theory of Cryptography*, vol. 10678. Heidelberg, Germany: Springer, 2017, pp. 489–518, doi: 10.1007/978-3-319-70503-3_16.

[7] F. Luo, F. Wang, K. Wang, and K. Chen, "A more efficient leveled strongly-unforgeable fully homomorphic signature scheme," *Inf. Sci.*, vol. 480, pp. 70–89, Apr. 2019, doi: 10.1016/j.ins.2018.12.025.

[8] Y. Wang and M. Wang, "A new fully homomorphic signatures from standard lattices," in *Proc. Int. Conf. Wireless Algorithms*, vol. 122384, 2020, pp. 494–506.

[9] R. Li, F. Wang, R. Zhang, and K. Chen, "NTRU-based fully homomorphic signature," *Secur. Commun. Netw.*, vol. 2022, pp. 1–11, Jun. 2022, doi: 10.1155/2022/9942717.

[10] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "Fully homomorphic encryption without bootstrapping," in *Proc. ITCS*, 2012, pp. 309–325.

[11] A. López-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption," in *Proc. 44th Annu. ACM Symp. Theory Comput.*, May 2012, pp. 1219–1234, doi: 10.1145/2213977.2214086.

[12] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical GapSVP," in *Advances in Cryptology—CRYPTO 2012*, vol. 7417. Heidelberg, Germany: Springer, 2012, pp. 868–886.

[13] J. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Advances in Cryptology—CRYPTO 2012*, vol. 10624. Heidelberg, Germany: Springer, 2017, pp. 409–437.

[14] C. Biswas and R. Dutta, "Secure and efficient multi-key FHE scheme supporting multi-bit messages from LWE preserving non-interactive decryption," *J. Ambient Intell. Humanized Comput.*, vol. 2022, pp. 1–14, May 2022, doi: 10.1007/s12652-022-03864-3.

[15] F. Wang, K. Wang, B. Li, and Y. Gao, "Leveled strongly-unforgeable identity-based fully homomorphic signatures," in *Proc. Int. Conf. Inf. Secur.*, in Lecture Notes in Computer Science, vol. 9290, 2015, pp. 42–60, doi: 10.1007/978-3-319-23318-5_3.

[16] C. Wang, B. Wu, and H. Yao, "Leveled adaptively strong-unforgeable identity-based fully homomorphic signatures," *IEEE Access*, vol. 8, pp. 119431–119447, 2020, doi: 10.1109/ACCESS.2020.3003685.

[17] Y. Zheng, "Digital signcryption or how to achieve cost(signature & encryption) $\ll$ cost(signature) + cost(encryption)," in *Advances in Cryptology—CRYPTO*, vol. 1294. Heidelberg, Germany: Springer, 1997, pp. 165–179, doi: 10.1007/BFb0052234.

[18] C. Gamage, J. Leiwo, and Y. Zheng, "Encrypted message authentication by firewalls," in *Proc. Int. Workshop Pract. Theory Public Key Cryptogr.*, in Lecture Notes in Computer Science, vol. 1560, 1999, p. 634, doi: 10.1007/3-540-49162-7_6.

[19] R. Steinfeld and Y. Zheng, "A signcryption scheme based on integer factorization," in *Proc. Int. Workshop Inf. Secur.*, in Lecture Notes in Computer Science, vol. 1975, 2000, pp. 308–322, doi: 10.1007/3-540-44456-4_23.

[20] J. Baek, R. Steinfeld, and Y. Zheng, "Formal proofs for the security of signcryption," in *Proc. Int. Workshop Public Key Cryptogr.*, vol. 2274, 2002, pp. 80–98, doi: 10.1007/3-540-45664-3_6.

[21] J. An, Y. Dodis, and T. Rabin, "On the security of joint signature and encryption," in *Advances in Cryptology—EUROCRYPT 2002*, in Lecture Notes in Computer Science, vol. 2332. Heidelberg, Germany: Springer, 2002, pp. 83–107, doi: 10.1007/3-540-46035-7_6.

[22] J. Malone-Lee, "Identity-based signcryption," *IACR Cryptol. ePrint Arch.*, vol. 2002, p. 98, Jan. 2002. [Online]. Available: https://eprint.iacr.org/2002/098

[23] M. Barbosa and P. Farshim, "Certificateless signcryption," in *Proc. ACM Symp. Inf., Comput. Commun. Secur.*, 2008, pp. 369–372.

[24] Z. Lui, "Secure certificateless signcryption scheme," *Appl. Res. Comput.*, vol. 30, no. 5, pp. 1533–1535, 2013.

[25] L. He and X. Peng, "Safe and efficient remote attestation protocol based on bilinear pairings signcryption," *J. Comput. Appl.*, vol. 33, no. 10, pp. 2854–2857, 2013.

[26] M. Qi, J. Chen, and D. He, "Signcryption scheme with public verifiability and forward security," *Appl. Res. Comput.*, vol. 31, no. 10, pp. 3093–3094, 2014, doi: 10.3969/j.issn.1001-3695.2014.10.051.

[27] X. Zhou, Z. Jin, and Y. Fu, "Efficient short signcryption from parings in Internet of Things," *J. Syst. Simul.*, vol. 26, no. 7, pp. 1566–1569, 2014.

[28] H. F. Yu and B. Yang, "Provably secure certificateless hybrid signcryption," *Chin. J. Comput.*, vol. 38, no. 4, pp. 804–813, 2015.

[29] J. Gan, X. Wu, and Y. Qin, "Secure certificateless signcryption scheme without bilinear pairing," *J. Softw.*, vol. 28, no. 10, pp. 2757–2768, 2017, doi: 10.13328/j.cnki.jos.005150.

[30] H. Wang and Y. Zhao, "Identity-based higncryption," *IACR Cryptol. ePrint Arch.*, vol. 2019, p. 106, Jan. 2019. [Online]. Available: https://eprint.iacr.org/2019/106

[31] M. Bellare and I. Steganovs, "Security under message-derived keys: Signcryption in iMessage," *Cryptol. ePrint Arch.*, vol. 2020, p. 224, Jan. 2020. [Online]. Available: https://eprint.iacr.org/2020/224

[32] Z. Liu, Y. Tseng, and R. Tso, "Cryptanalysis on 'an efficient identity-based proxy signcryption using lattice,'" *Cryptol. ePrint Arch.*, vol. 2021, p. 359, Jan. 2021. [Online]. Available: https://eprint.iacr.org/2021/359

[33] S. Hu, R. Zhang, F. Wang, K. Chen, B. Lian, and G. Chen, "A sanitizable signcryption scheme with public verifiability via chameleon hash function," *J. Inf. Secur. Appl.*, vol. 71, Dec. 2022, Art. no. 103371, doi: 10.1016/j.jisa.2022.103371.

[34] M. O. S. Ajtai, "Generating hard instances of lattice problems (extend abstract)," in *Proc. STOC*, G. L. Miller, Ed. 1996, pp. 409–437.

[35] D. Micciancio, "Almost perfect lattices, the covering radius problem, and applications to Ajtai's connection factor," *SIAM J. Comput.*, vol. 34, no. 1, pp. 118–169, Jan. 2004, doi: 10.1137/S0097539703433511.

[36] D. Micciancio and O. Regev, "Worst-case to average-case reductions based on Gaussian measures," *SIAM J. Comput.*, vol. 37, no. 1, pp. 267–302, 2007, doi: 10.1137/S0097539705447360.

[37] D. Micciancio and C. Peikert, "Hardness of SIS and LWE with small parameters," in *Advances in Cryptology—CRYPTO 2013*, vol. 8042. Heidelberg, Germany: Springer, 2013, pp. 21–39.

[38] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *J. ACM*, vol. 56, no. 6, pp. 1–40, 2009, doi: 10.1145/1568318.1568324.

[39] M. Ajtai, "Generating hard instances of the short basis problem," in *Proc. ICALP*, J. Wiedermann, P. Boas, and M. Nielsen, Eds., vol. 1644, 1999, pp. 1–9, doi: 10.1007/3-540-48523-6_1.

[40] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proc. STOC*, C. Dwork, Ed., 2008, pp. 197–206.

[41] J. Alwen and C. Peikert, "Generating shorter bases for hard random lattices," *Theory Comput. Syst.*, vol. 48, pp. 535–553, Jul. 2011, doi: 10.1007/s00224-010-9278-3.

[42] D. Micciancio and C. Peikert, "Trapdoors for lattice: Simpler, tighter, faster, smaller," in *Advances in Cryptology—EUROCRYPT 2012*, vol. 7237. Heidelberg, Germany: Springer, 2012, pp. 700–718. [Online]. Available: http://www.iacr.org/workshops/ches/ches2010


**XIAODAN JIN** received the B.S. degree in mathematics and applied mathematics from Chizhou University, Chizhou, Anhui, China, in 2020. She is currently pursuing the M.S. degree with the School of Mathematics, Hangzhou Normal University. Her research interests include fully homomorphic cryptography and lattice-based cryptography.

**FUQUN WANG** received the B.S. degree in information and computing science and the M.S. degree in pure mathematics from Zhengzhou University, Zhengzhou, Henan, China, in 2003 and 2006, respectively, and the Ph.D. degree in information security from the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China, in 2016. He is currently an Associate Professor with the Department of Mathematics, Hangzhou Normal University. His research interests include fully homomorphic cryptography and lattices-based cryptography.

**RENJUN ZHANG** received the Ph.D. degree from the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China, in 2019. He is currently an Assistant Professor with the School of Mathematics, Hangzhou Normal University, Hangzhou, China. His research interests include public-key cryptography and blockchain.

**BIN LIAN** received the M.S. degree in cryptography from Southwest Jiaotong University, in 2005, and the Ph.D. degree in cryptography from Shanghai Jiao Tong University, in 2015. He is currently an Associate Professor with NingboTech University, Ningbo, China. His research interests include cryptography, cryptographic protocol, and the technology of network security.

**KEFEI CHEN** (Member, IEEE) received the B.S. and M.S. degrees in applied mathematics from Xidian University, Xi'an, in 1982 and 1985, respectively, and the Ph.D. degree from Justus-Liebig University Giessen, Germany, in 1994. From 1996 to 2012, he was a Professor with the Department of Computer Science and Engineering, Shanghai Jiaotong University. He is currently a Professor with the Department of Mathematics, Hangzhou Normal University. His research interests include cryptography, theory, and the technology of network security.