Hindawi
IET Information Security
Volume 2023, Article ID 1377042, 8 pages
https://doi.org/10.1049/2023/1377042

Research Article

A Publicly Verifiable Leveled Fully Homomorphic Signcryption Scheme

Zhaoxuan Bian ,[1,2] Fuqun Wang ,[1,2] Renjun Zhang ,[1,2] Bin Lian,[3] Lidong Han,[2] and Kefei Chen[1,2]

[1] *School of Mathematics, Hangzhou Normal University, Hangzhou 31121, China*
[2] *Key Laboratory of Cryptography of Zhejiang Province, Hangzhou Normal University, Hangzhou 31121, China*
[3] *NingboTech University, Ningbo 315199, China*

Correspondence should be addressed to Fuqun Wang; fqwang@hznu.edu.cn and Renjun Zhang; zhangrenjun@hznu.edu.cn

Received 13 August 2023; Revised 24 September 2023; Accepted 28 September 2023; Published 31 October 2023

Academic Editor: Guowen Xu

With the deepening of research, how to construct a fully homomorphic signcryption scheme based on standard assumptions is a problem that we need to solve. For this question, recently, Jin et al. proposed a leveled fully homomorphic signcryption scheme from standard lattices. However, when verifying, it is supposed to unsigncrypt first as they utilize sign-then-encrypt method. This leads to users being unable to verify the authenticity of the data first, which resulting in the waste of resources. This raises another question of how to construct an fully homomorphic signcryption (FHSC) scheme with public verifiability. To solve this problem, we propose a leveled fully homomorphic signcryption scheme that can be publicly verified and show its completeness, IND-CPA security, and strong unforgeability.

1. Introduction

With the rapid development of the digital economy, data have become an extremely important social resource. In practical applications, we often need to rely on third-party computing power to help us perform calculations on data. In this process, how to ensure the privacy and authentication of data have become a problem that we need to solve.

The groundbreaking development of fully homomorphic encryption (FHE) by Gentry [1] makes it possible for server to homomorphically perform arbitrary computations over the ciphertexts. Inspired by Gentry's work, many FHE schemes have emerged subsequently such as in studies by Brakerskiand Vaikuntanathan [2], Brakerski [3], Brakerski et al. [4], Gentry et al. [5], and Cheon et al. [6]. FHE helped us achieve homomorphic operations on ciphertext, but it cannot provide the verifiability of the data.

In the terms of its duality problem, Gorbunov et al. [7] proposed the first leveled fully homomorphic signature (FHS) schemes based on the short integer solution (SIS) problem in standard lattices and come up with a new primitive named homomorphic trapdoor function (HTDF) in 2015. Given a set of messages $\vec{\mu}$, corresponding signatures $\vec{\sigma}$, and a function f, FHS allows third party to obtain the signature $f(\vec{\sigma})$ of plaintext $f(\vec{\mu})$ through homomorphic computation. In the same year, Wang et al. [8] devised a leveled identity-based FHS scheme with strong unforgeability. In their paper, they first extended the notion of HTDF and obtained the identity-based HTDF, which has better parameters and stronger security. Li et al. [9] established an FHS scheme based on NTRU and provided a new content for this field in 2022. Unfortunately, FHS only guarantees the authenticity of the data, while the private data itself remain exposed.

In many situations, we need to simultaneously realize privacy and authenticity. In 1997, Zheng [10] proposed a new cryptographic primitive named digital signcryptation to balance the privacy and authenticity of data. However, their scheme does not support homomorphic operations. In 2017, Rezaeibagha et al. [11] proposed a homomorphic signcryption (HSC) scheme based on the decisional Diffie–Hellman assumption which only supported linear homomorphic operations. Furthermore, Li et al. [12] constructed two leveled fully homomorphic signcryption (FHSC) schemes under nonstandard assumptions as they used indistinguishable obfusion, zero knowledge proof, and multiple input function encryption. Recently, Jin et al. [13] proposed a leveled FHSC schemes based on lattices. However, it cannot be publicly verified, as it needs to decrypt the ciphertext first and then verify, which resulting in the waste of resources. How to construct an FHSC scheme with public verifiability is a problem that needs to be addressed. In this paper, we provide a positive answer.

1.1. Contribution. To solve the problem of how to construct an FHSC scheme with public verifiability, we propose a publicly verifiable leveled FHSC scheme, which is more practical than Jin's scheme. For this purpose, we extend the encrypt-then-sign method from signcryption setting to FHSC setting to achieve the function of public verification. Due to the method, we don't need to decrypt the ciphertext first before verifying. In other words, our FHSC supports public verifiability. Furthermore, given a set of encrypted data, our scheme can achieve fast verification through the homomorphism. Additionally, we show the completeness, strong unforgeability, and IND-CPA security of our scheme.

1.2. Organization. The structure and basic content of this article are as follows. First, we describe some background on lattice, related homomorphic schemes, and some definations related to FHSC in Section 1. Second, we provide our scheme construction, homomorphic operation, noise analysis, and security in Section 3. Finally, we conclude our paper in Section 4.

2. Preliminaries

2.1. Basic Notion. In this paper, we denote the ring of integers as $\mathbb{Z}$. We use lowercase bold letters, e.g., $\mathbf{x}$ to represent vectors and capital letters, e.g., X to denote matrices. Given a distribution χ, the formula $x \xleftarrow{\$} \chi$ denotes the process that sample x from χ uniformly at random. In addition, we denote the infinite norm of A as $\|A\|_\infty$. Throughout, we denote the security parameter as λ and denote negligible functions as $negl(\lambda)$.

2.2. Background on the Lattices. Let $A = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n)$ for each $\mathbf{a}_{i \in [n]} \in \mathbb{R}^n$ be a set of linearly independent basis vectors. We say that $\Lambda = \mathscr{L}(A) = \{Az = \sum_{i=1}^n z_i \cdot \mathbf{a}_i : z_i \in \mathbb{Z}\}$ is a lattice generated by A.

Definition 1 ([14] (short integer solution)). Let $n = n(\lambda)$, $m = m(\lambda), q = q(\lambda), \beta = \beta(\lambda) > 0$ be integer parameters defined in terms of the security parameter λ. Given a matrix $A \in Z_q^{n \times m}$ consists of m vectors $\mathbf{a}_i \in \mathbb{Z}_q^n$ selected uniformly at random. In the SIS problem, the adversary wants to find a small vector $\mathbf{t} \in \mathbb{Z}^m$ satisfying $\mathbf{t} \neq \mathbf{0}$ and $\|\mathbf{t}\|_\infty \leq \beta$ such that $A\mathbf{t} = \mathbf{0}$, and the SIS problems can be reduced to certain worst-case problems in the standard lattices [15–18].

Definition 2 ([5] (α-bounded distribution)). A set of distributions $\{\chi_n\}_{n \in N}$ supported over the integers is α-bounded if the distribution satisfies $\Pr_{\epsilon \leftarrow \chi_n} [|\epsilon| > \alpha] = Negl(n)$.

Definition 3 ([19] (learning with error (LWE))). Given positive integers n, m, q, and χ which is a distribution over $\mathbb{Z}_q$. The LWE problem is to find a vector $\mathbf{s}$ which satisfies $(A, \langle A, \mathbf{s}\rangle + \mathbf{e})$ over $\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$, where A consists of $\mathbf{a}_i \xleftarrow{\$} \mathbb{Z}_q^m, i \in [n]$, $\mathbf{e} \leftarrow \chi^m$ and $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$. The LWE assumption is that the LWE problem is infeasible.

Lemma 4. *[7, 20–23] There exist a tuple of efficient algorithm consists of* $\mathsf{TrapGen}, \mathsf{SamplePre}, \mathsf{Sample}$ *such that the following holds. Given positive integers $n > 1, q \geq 2$, we can obtain the following relationships about $m^*(n, q)$ and $\beta_{sam}(n, q)$ for all $m \geq m^*$ and all $k = poly(n)$:*

(1) $U \leftarrow \mathsf{Sample}(1^m, 1^k, q)$: *We sample a matrix $U \in \mathbb{Z}_q^{m \times k}$ which satisfies $\|U\|_\infty \leq \beta_{sam}$*

(2) *The following two distribution statistics are indistinguishable: $A \approx_S A^*$ and $(A, Td, U, V) \approx_S (A, Td, U^*, V^*)$, where*

$$\begin{cases} (A, Td) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q) \\ A^* \xleftarrow{\$} \mathbb{Z}_q^{n \times m} \\ U \leftarrow \mathsf{Sample}\left(1^m, 1^k, q\right) \\ V \triangleq A \cdot U \\ U^* \leftarrow \mathsf{SamplePre}(A, V^*, Td) \\ V^* \xleftarrow{\$} \mathbb{Z}_q^{n \times k}. \end{cases} \tag{1}$$

Moreover, any $U^* \in \mathrm{SamplePre}(A, V^*, Td)$ always satisfies $AU^* = V^*$ and $\|U^*\|_\infty \leq \beta_{sam}$. From this, it can be concluded that the statistical distances are negligible in λ.

(3) *Received n, m, q as above, there is a deterministic matrix $G \in \mathbb{Z}_q^{n \times m}$ and a deterministic algorithm G^{-1} that can be effectively calculated. For a $V \in \mathbb{Z}_q^{n \times m}$, we can obtain $\widehat{V} = G^{-1}(V)$ where $\widehat{V} \in \{0, 1\}^{m \times m}$ and $G \cdot \widehat{V} = V$.*

Next, we will introduce two homomorphic schemes that play important roles in our scheme.

2.3. Associated Homomorphic Schemes

2.3.1. GSW-FHE [5]. A GSW-FHE scheme consists of a tuple of algorithms $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Evaluate})$ as follows:

(i) $GSW.prms \leftarrow \mathsf{GSW.Setup}(1^\lambda, 1^L)$: Input security parameter λ, maximum homomorphic depth L, and output $GSW.prms = (n_1, m_1, q_1, N_1)$, where $N_1 = (n_1 + 1)\log q$.

(ii) $(pk, sk) \leftarrow \mathsf{KeyGen}(GSW.prms)$: Take the $GSW.prms$ as input and samples $\mathbf{t} \leftarrow \mathbb{Z}_{q_1}^{n_1}$. Then, generate a matrix $D \leftarrow \mathbb{Z}_{q_1}^{m_1 \times n_1}$ uniformly and a vector $\mathbf{e} \leftarrow \chi^{m_1}$. Set $\mathbf{b} = D\mathbf{t} + \mathbf{e}$ and $B = (\mathbf{b}, D)$. Then, we can obtain the public key $pk = B$ (Remark: Observe that $B\mathbf{s} = \mathbf{e}$). Output $pk = B, sk = (1, -\mathbf{t}) \in \mathbb{Z}_{q_1}^{n_1+1}$.

(iii) $C \leftarrow \mathsf{Enc}(GSW.prms, m, pk)$: Input the public parameter $GSW.prms$, a message $\mu \in \mathscr{M}$, and public key pk. Then, output $C = Flatten(Bitdecomp(RB) + \mu I_N) \in \mathbb{Z}_{q_1}^{N_1 \times N_1}$ where $R \xleftarrow{\$} \{0, 1\}^{N_1 \times m_1}$.

(iv) $\mu \leftarrow \mathsf{Dec}(GSW.prms, C, sk)$: Compute $\langle C, Powersof2(\mathbf{s}) \rangle = \mu Powersof2(\mathbf{s}) + \mathrm{Re}$.

(v) $C^* \leftarrow \mathsf{Evaluate}(GSW.prms, C_1, C_2, \ldots, C_L, f)$: Input $(C_1, C_2, \ldots, C_L, f)$ and output C^*.

2.3.2. GVW-FHS [7]. A GVW-FHS scheme consists of a tuple of algorithms ($\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Sign\text{-}Eval}, \mathsf{Process}, \mathsf{Verify}$) as follows:

(i) $GVW.prms \leftarrow \mathsf{GVW.Setup}(1^\lambda, 1^{N_2})$: It takes $(V_1, V_2, \ldots, V_{N_2})$ by sample $V_i \xleftarrow{\$} \mathscr{V}$ as the input and generates parameters (n_2, m_2, q_2). We record all generated parameters as $GVW.prms = (V_1, V_2, \ldots, V_{N_2}, n_2, m_2, q_2)$.

(ii) $(pk, sk) \leftarrow \mathsf{KeyGen}(GVW.prms)$: It outputs $(A, Td) \leftarrow \mathrm{TrapGen}(1^{n_2}, 1^{m_2}, q_2)$ and denotes as $pk = A$, $sk = Td$.

(iii) $U \leftarrow \mathsf{Sign}(GVW.prms, \mu, sk)$: Input the data x to be signed, $GVW.prms$, the secret key sk, and out the signature U.

(iv) $U^* \leftarrow \mathsf{SignEval}(f, (\mu_1, V_1, U_1), (\mu_2, V_2, U_2), \ldots, (\mu_{N_2}, V_{N_2}, U_{N_2}), pk)$: Input $(f, (\mu_1, V_1, U_1), (\mu_2, V_2, U_2), \ldots, (\mu_{N_2}, V_{N_2}, U_{N_2})$ and output U^*.

(v) $V_f \leftarrow \mathsf{Process}(GVW.prms, f)$: Input $(f, V_1, V_2, \ldots, V_{N_2})$. And output V_f.

(vi) $0/1 \leftarrow \mathsf{Verify}(GVW.prms, pk, U^*, \mu^*, f)$: If $f_{(pk,y)}(U^*) = V^*$, where $y = (f, \mu_1, \mu_2, \ldots, \mu_{N_2})$, then output 1. Otherwise output 0.

Lemma 5. *Based on SIS problem, which is considered difficult, the GVW-FHS scheme [7] satisfies existential unforgeability. Furthermore, we can obtain that the GVW-FHS scheme is strongly-unforgeable adapted from the identity-based FHS scheme [8].*

2.4. Definitions Related to FHSC. In this section, we will describe the commonly known definitions for FHSC scheme, as well as the completeness, IND-CPA security, and strong unforgeability.

2.4.1. FHSC. A fully homomorphic signcryption scheme is a tuple of algorithms consisting of ($\mathsf{Setup}, \mathsf{KeyGen}_s, \mathsf{KeyGen}_r, \mathsf{Signcrypt}, \mathsf{Unsigncrypt}, \mathsf{Eval}, \mathsf{Process}, \mathsf{Verify}$) as follows:

(i) $prms \leftarrow \mathsf{Setup}(1^\lambda, 1^L, 1^S)$: Get the λ, maximum homomorphic depth L, and a data-size bound S. Then, output the public parameter $prms$ and the message space $\mathscr{M}$.

(ii) $(pk_s, sk_s) \leftarrow \mathsf{KeyGen}_s(prms)$: Input the $prms$ and generate the sender's key pair (pk_s, sk_s).

(iii) $(pk_r, sk_r) \leftarrow \mathsf{KeyGen}_r(prms)$: Input the $prms$ and generate the receiver's key pair (pk_r, sk_r).

(iv) $\sigma \leftarrow \mathsf{Signcrypt}(prms, \mu, sk_s, pk_r)$: Input public parameter $prms$, a message $\mu \in \mathscr{M}$, sender's private key sk_s and receiver's public key pk_r, and output a signcryption σ.

(v) $\mu \leftarrow \mathsf{Unsigncrypt}(prms, \sigma, pk_s, sk_r)$: Input public parameter $prms$, the signcryption σ, sender's public key pk_s and receiver's private key sk_r, and output μ after verifing the integrity of ciphertext.

(vi) $\sigma* \leftarrow \mathsf{Eval}(prms, pk_s, pk_r, f, \sigma_1 \ldots, \sigma_S)$: Input $\sigma_1 \ldots, \sigma_S$, and output homomorphic signcryption $\sigma*$.

(vii) $V_f \leftarrow \mathsf{Process}(prms, f)$: Input public parameter $prms$ and function f. Homomorphically computes a V_f, which is used for verification.

(viii) $0/1 \leftarrow \mathsf{Verify}(prms, pk_s, \sigma^*, f)$: Input the evaluated signcrtption σ^*, sender's public key pk_s, and output 0/1.

2.4.2. Completeness. Given messages $(\mu_1, \mu_2, \ldots, \mu_S) \in \mathscr{M}$, $f(\mu_1, \mu_2, \ldots, \mu_S) = \mu^*$, $prms$, (pk_s, sk_s), and (pk_r, sk_r), we can obtain the signcryption σ_i of each message μ_i and the homomorphic operation result σ^*. It satisfies the following properties with a nonnegligible probability:

$$\begin{cases} 1 \ = \mathsf{Verify}(prms, pk_s, \sigma^*, f) \\ \mu^* = \mathsf{Unsigncrypt}(prms, \sigma^*, sk_r) \end{cases} \quad (2)$$

2.4.3. IND-CPA Security. We say that an FHSC scheme satisfies IND-CPA security if and only if a probabilistic polynomial time (PPT) adversary $\mathscr{A}$ has a negligible advantage to win the following game.

(1) The challenger $\mathscr{C}$ first obtain the $prms$ and the key pair (pk_r, sk_r) from the Setup and KeyGen_r. Then, $\mathscr{C}$ sends the $(pk_r, sk_s, prms)$ to $\mathscr{A}$.

(2) $\mathscr{A}$ chooses two plaintexts μ_0, μ_1, satisfing $|\mu_0| = |\mu_1|$ and then run the KeyGen_s to get the (pk_s, sk_s). Finally, $\mathscr{A}$ gives the $(\mu_0, \mu_1, pk_s, sk_s)$ to $\mathscr{C}$.

(3) $\mathscr{C}$ chooses a random bit $b \leftarrow \{0, 1\}$ and sents the σ_b to $\mathscr{A}$ where $\sigma_b \leftarrow \mathsf{Signcrypt}(prms, \mu_b, sk_s, pk_r)$.

(4) $\mathscr{A}$ outputs a bit $b' \leftarrow \{0, 1\}$. If $b' = b$, $\mathscr{A}$ wins.

The advertange of the adversary to win the game is:

$$\mathrm{Adv}_{\mathscr{A}}^{\mathrm{IND-CPA}} = |Pr[b' = b] - 1/2|. \quad (3)$$

2.4.4. Strong Unforgeability. We say that an FHSC scheme satisfies strong unforgeability under chosen message attack if there is no PPT forger $\mathscr{F}$ can win the following game with a nonnegligible advantage.

(1) The challenger $\mathscr{C}$ first generates the *prms* and the key pair (pk_r, pk_s, sk_r, sk_s) from the Setup, KeyGen_s, and KeyGen_r. Then, $\mathscr{C}$ sends the $(pk_r, pk_s, prms)$ to $\mathscr{F}$.
(2) $\mathscr{F}$ chooses and sends plaintexts $(\mu_1, \mu_2, \ldots, \mu_S)$ to $\mathscr{C}$.
(3) $\mathscr{C}$ obtains the σ_i from $\mathsf{Signcrypt}$ and sents the $(\sigma_1, \sigma_2, \ldots, \sigma_S)$, where $i \in [S]$ to $\mathscr{F}$.
(4) $\mathscr{F}$ chooses and sends a function $f \in F$, as well as a value σ' to $\mathscr{C}$.
(5) $\mathscr{F}$ wins if all of the following hold:
f is admissible on the messages $\mu_1, \mu_2, \ldots, \mu_S$;
$\sigma_f \neq \sigma'$, where $\sigma_f = f(\sigma_1, \sigma_2, \ldots, \sigma_S)$;
$\mathsf{Verify}(\sigma', V_f)$ accept, where $V_f = \mathsf{Process}(prms, f)$.

An FHSC scheme is SU-CMA security if:

$$\left|\Pr\left[\mathrm{Exp}^{\mathrm{SU-CMA}}_{\mathscr{F},\mathrm{FHSC}}(1^\lambda)\right]\right| < \mathrm{negl}(\lambda). \tag{4}$$

Remark 6. Remark that we do not require either $\mu' = f(\mu_1, \mu_2, \ldots, \mu_S)$ or not. So, if $\mu' = f(\mu_1, \mu_2, \ldots, \mu_S)$, then σ' is a valid signcryption to break the strong unforgeability of FHSC scheme, otherwise a valid signcryption to break the existent unforgeability.

3. The Proposed FHSC Scheme

In this section, we will represent our FHSC construction, homomorphic operation, noise analysis, and security.

3.1. Basic Construction. Our scheme will be defined by a flexible parameter $L = L(\lambda) = poly(\lambda)$ which the depth of homomorphism. We choose parameters: $n, m, q, \beta_{SIS}, \beta_{\max}$, β_{init} depending on λ and L. We do so by setting $\beta_{\max} \triangleq 2^{\omega(\log\lambda)L}$ and $\beta_{SIS} \triangleq 2^{\omega(\log\lambda)L}\beta_{\max}$. Then, we let $n = poly(\lambda)$ and prime $q = 2^{poly(\lambda)} > \beta_{SIS}$ be integers as small as possible to ensure that the $SIS(n, m, q, \beta_{SIS})$ assumption holds for all $m = poly(\lambda)$. In the end, we denote $m^* = m^*(n, q) \triangleq O(n\log q), \beta_{sam} = O\left(n\sqrt{\log q}\right)$ as the parameters required by the algorithms TrapGen, as shown in Lemma 4, and set $m = \max\{m, n\log q + \omega(\log\lambda)\} = poly(\lambda)$ while $\beta_{init} \triangleq \beta_{sam} = poly(\lambda)$. Note that $n, m, \log q$ all depend on poly(L,λ).

Our leveled FHSC scheme consists of polytime algorithms (Setup, KeyGen_s, $\mathsf{K\text{-}eyGen}_r$, $\mathsf{Signcrypt}$, $\mathsf{Unsigncrypt}$, Eval, $\mathsf{Process}$, Verify) with syntax:

(i) $prms \leftarrow \mathsf{Setup}(1^\lambda, 1^L, 1^S)$. Input the security parameter λ, homomorphic depth L, and a data-size bound S. Then, run $GSW.prms \leftarrow \mathsf{PrmsGen}(1^\lambda, 1^L)$ and $GVW.prms \leftarrow \mathsf{PrmsGen}(1^\lambda, 1^S)$, where $GSW.prms = (n_1, m_1, q_1, \chi)$ and $GVW.prms = (\mathscr{V}_1, \mathscr{V}_2, \ldots, \mathscr{V}_S, n_2, m_2, q_2)$. Remark that $\mathscr{V}_{i\in[S]}$ consists of N^2 matrices, each of which has n rows and m columns where $m = \max\{m_1, m_2\}$ and $n = \max\{n_1, n_2\}$. Let $l = \lceil\log q\rceil, N = (n+1)l, q = \max\{q_1, q_2\}$, domains $\mathscr{M} = \mathbb{Z}_q$, and $\mathscr{V} = \mathbb{Z}_q^{nN\times mN}$. Define the distruibution $D_{\mathscr{U}}$ to sample $\mathscr{U} \leftarrow \mathsf{Sample}(1^{mN}, 1^{mN}, q)$, as shown in Lemma 4, which satisfies $\|\mathscr{U}\|_\infty \leq \beta_{init}$. Let $\mathscr{U} = \left(U_{ij} \in \mathbb{Z}_q^{m\times m}\right)_{i,j\in[N]}$. Finally, output $prms = (\mathscr{V}_1, \mathscr{V}_2, \ldots, \mathscr{V}_S, n, m, q, \chi)$.
(ii) $(pk_s, sk_s) \leftarrow KeyGen_s(prms)$. Run $(A, td) \leftarrow \mathsf{GVW.KeyGen}_s(prms)$ and set $pk_s = A \in \mathbb{Z}_q^{n\times m}$, $sk_s = td \in \mathbb{Z}_q^m$.
(iii) $(pk_r, sk_r) \leftarrow KeyGen_r(prms)$. Run $(B, \mathbf{s}) \leftarrow \mathsf{GSW.KeyGen}_r(prms)$ and set $pk_r = B \in \mathbb{Z}^{m\times(n+1)}$, $sk_r = \mathbf{s}$, where $s = (1, -\mathbf{t}) \in \mathbb{Z}_q^{n+1}$, $B = (\mathbf{b}, D) \in \mathbb{Z}_q^{m\times(n+1)}$ and $\mathbf{s}B = \mathbf{e}$.
(iv) $(C, U) \leftarrow \mathsf{Signcrypt}(prms, \mu, sk_s, pk_r)$.

(1) For a message $\mu \in \mathscr{M}$, run $C \leftarrow \mathsf{GSW.Enc}(\mu, pk_r)$. Remark that $C = Flatten(Bitdecomp(RB) + \mu I_N)$, where $R \xleftarrow{\$} \{0, 1\}^{N\times m}$. Let

$$C = \begin{pmatrix} C_{00} & C_{01} & \cdots & C_{0N-1} \\ C_{10} & C_{11} & \cdots & C_{1N-1} \\ \vdots & \vdots & \ddots & \vdots \\ C_{N-10} & C_{N-10} & \cdots & C_{N-1N-1} \end{pmatrix} \tag{5}$$

be a matrix whose entries consist of {0, 1}

(2) For $\forall C_{ij}$, run $U_{ij} \leftarrow GVW.Sign(prms, C_{ij}, sk_s)$. Remark that $V_{ij} = AU_{ij} + C_{ij}G$. Let

$$\mathscr{U} = \begin{pmatrix} U_{00} & U_{01} & \cdots & U_{0N-1} \\ U_{10} & U_{11} & \cdots & U_{1N-1} \\ \vdots & \vdots & \ddots & \vdots \\ U_{N-10} & U_{N-10} & \cdots & U_{N-1N-1} \end{pmatrix} \tag{6}$$

and

$$\begin{aligned}\mathscr{V} &= \begin{pmatrix} V_{00} & V_{01} & \dots & V_{0N-1} \\ V_{10} & V_{11} & \dots & V_{1N-1} \\ \vdots & \vdots & \ddots & \vdots \\ V_{N-10} & V_{N-10} & \dots & V_{N-1N-1} \end{pmatrix} \\ &= \begin{pmatrix} AU_{00}+C_{00}G & AU_{01}+C_{01}G & \dots & AU_{0n}+C_{0N-1}G \\ AU_{10}+C_{10}G & AU_{11}+C_{11}G & \dots & AU_{1N-1}+C_{1N-1}G \\ \vdots & \vdots & \ddots & \vdots \\ AU_{N-10}+C_{N-10}G & AU_{N-10}+C_{N-10}G & \dots & AU_{N-1N-1}+C_{N-1N-1}G \end{pmatrix} \\ &= A\mathscr{U}+CG, \end{aligned} \tag{7}$$

where the matrixes A and G are seen as a number in the multiplication operation, respectively.

(i) $\mu \leftarrow \mathsf{Uncryptsign}(prms, C, \mathscr{U}, pk_s, sk_r)$. Input the public parameter $prms$ and signcryption$(C, \mathscr{U})$. Run algorithm $0/1 \leftarrow \mathsf{Verify}(C, \mathscr{U})$ first and then run algorithm $\mu \leftarrow \mathsf{GSW.Dec}(prms, C, sk_r)$ if $1 \leftarrow \mathsf{Verify}(C, \mathscr{U})$.

(ii) $(C^*, \mathscr{U}^*) \leftarrow \mathsf{Eval}(prms, f, C_1, C_2, \dots, C_S,)$. Input $(prms, pk_s, pk_r, f)$, as well as $(C_1, \mathscr{U}_1), (C_2, \mathscr{U}_2), \dots, (C_S, \mathscr{U}_S)$, and output the homomorphic signcryption $(C^*, \mathscr{U}^*)$.

(iii) $\mathscr{V}_f \leftarrow \mathsf{Process}(prms, f)$. Takes the admissible function f and $prms$ as inputs. Then, output $\mathscr{V}_f \leftarrow \mathsf{GVW.Process}(prms, f)$.

(iv) $0/1 \leftarrow \mathsf{Verify}(\mathscr{V}_f, pk_s, C_f, \mathscr{U}_f)$. Take the $\mathscr{V}_f$, sender's public key pk_s, and the signcryption$(C_f, \mathscr{U}_f)$ as input. Then output 1 if $\mathscr{V}_f = A\mathscr{U}_f + C_f G$, otherwise output 0.

(v) $\mu_f \leftarrow \mathsf{GSW.Dec}(prms, sk_r, C_f)$. Input public parameter $prms$ and secret key sk_r and output the new message under f if $1 \leftarrow \mathsf{Verify}(prms, pk_s, C_f, \mathscr{U}_f, f)$.

3.2. Homomorphic Evalution and Noise Analysis. Here, we describe the additive homomorphism and multiplicative homomorphism. For

$$C_1 = \begin{pmatrix} C_{00} & C_{01} & \dots & C_{0N-1} \\ C_{10} & C_{11} & \dots & C_{1N-1} \\ \vdots & \vdots & \ddots & \vdots \\ C_{N-10} & C_{N-10} & \dots & C_{N-1N-1} \end{pmatrix},$$

$$C_2 = \begin{pmatrix} \widehat{C}_{00} & \widehat{C}_{01} & \dots & \widehat{C}_{0N-1} \\ \widehat{C}_{10} & \widehat{C}_{11} & \dots & \widehat{C}_{1N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \widehat{C}_{N-10} & \widehat{C}_{N-10} & \dots & \widehat{C}_{N-1N-1} \end{pmatrix} \text{ and}$$

$$\mathscr{U}_1 = \begin{pmatrix} U_{00} & U_{01} & \dots & U_{0n} \\ U_{10} & U_{11} & \dots & U_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ U_{N-10} & U_{N-10} & \dots & U_{N-1n} \end{pmatrix},$$

$$\mathscr{U}_2 = \begin{pmatrix} \widehat{U}_{00} & \widehat{U}_{01} & \dots & \widehat{U}_{0n} \\ \widehat{U}_{10} & \widehat{U}_{11} & \dots & \widehat{U}_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ \widehat{U}_{N-10} & \widehat{U}_{N-10} & \dots & \widehat{U}_{N-1n} \end{pmatrix}, \text{ while}$$

$$\mathscr{V}_1 = \begin{pmatrix} V_{00} & V_{01} & \dots & V_{0N-1} \\ V_{10} & V_{11} & \dots & V_{1N-1} \\ \vdots & \vdots & \ddots & \vdots \\ V_{N-10} & V_{N-10} & \dots & V_{N-1N-1} \end{pmatrix},$$

$$\mathscr{V}_2 = \begin{pmatrix} \widehat{V}_{00} & \widehat{V}_{01} & \dots & \widehat{V}_{0N-1} \\ \widehat{V}_{10} & \widehat{V}_{11} & \dots & \widehat{V}_{1N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \widehat{V}_{N-10} & \widehat{V}_{N-10} & \dots & \widehat{V}_{N-1N-1} \end{pmatrix}$$

Additive Homomorphism. We define that

$$\begin{cases} C_{\text{Add}} = \left(C_{ij} + \widehat{C}_{ij}\right)_{i,j\in[N]} \\ \mathscr{U}_{\text{Add}} = \left(U_{ij} + \widehat{U}_{ij}\right)_{i,j\in[N]} \\ \mathscr{V}_{\text{Add}} = \left(V_{ij} + \widehat{V}_{ij}\right)_{i,j\in[N]} = \left(A\left(U_{ij} + \widehat{U}_{ij}\right) + \left(C_{ij} + \widehat{C}_{ij}\right)G\right)_{i,j\in[N]} \end{cases} \tag{8}$$

For simplicity, $\mathscr{V}_{\text{Add}} = A\mathscr{U}_{\text{Add}} + C_{\text{Add}}G$, where A and G should be seen as a number while performing the multiplication operation.

For

$$\begin{aligned} C_1 &= \text{Flatten}(\text{Bitdecomp}(R_1B) + \mu I_N), \\ C_2 &= \text{Flatten}(\text{Bitdecomp}(R_2B) + \mu I_N), \end{aligned} \tag{9}$$

We can easily recover the μ_{Add} from the formula:

$$\begin{aligned} < C_{\text{Add}}, \text{Power of } 2(\text{sk}_r) >= {} & R_1e_1 + R_2e_2 \\ & + (\mu_1 + \mu_2) \text{ Powers of } 2(\text{sk}_r). \end{aligned} \tag{10}$$

Next, we analyze the noise variations during the additive homomorphic processes.

If the upper noise boundary of U_1 and U_2 is β, then we can easily obtain that the upper noise boundary of U_{Add} is 2β.

If the upper noise boundary of R_1e_1 and R_2e_2 is α, then we can easily obtain that the upper noise boundary of C_{Add} is 2α.

MultConst Homomorphism. We define that:

$$\begin{cases} C_{\mathrm{conMult}} = \mathrm{Flatten}(aI_N)C \\ \mathscr{U}_{\mathrm{Multconst}} = \mathrm{Flatten}(aI_N)\mathscr{U} \\ \mathscr{V}_{\mathrm{Multconst}} = \mathrm{Flatten}(aI_N)A\mathscr{U} + \mathrm{Flatten}(aI_N)CG. \end{cases} \tag{11}$$

From Equation (2), we can learn that:

$$\begin{aligned} < C_{\mathrm{Multconst}}, \text{ Powers of } 2(\mathrm{sk}_r) > &= a\mu \text{ Power of } 2(sk_r) \\ &\quad + \mathrm{Flatten}(aI_N)Re_2 \\ \left\| \mathscr{U}_{\mathrm{Multconst.ij}} \right\|_\infty &= a\left\| U_{ij} \right\|_\infty. \end{aligned} \tag{12}$$

If Re_1 is bounded by α, then the upper noise boundary of $C_{\mathrm{Multconst}}$ is $(N+1)\alpha$.

If U is bounded by β, then the upper noise boundary of $\mathscr{U}_{\mathrm{Multconst}}$ is $a\beta$.

Multiplicative Homomorphism. We define that:

$$\begin{cases} C_{\mathrm{Mult}} = C_2C_1 = \left(\sum_{k=0}^{N-1} \widehat{C}_{ik} C_{kj} \right)_{i,j\in[N]} \\ \mathscr{U}_{\mathrm{Mult}} = \mathscr{U}_2 \otimes \mathscr{U}_1 = \left(\left(\sum_{k=0}^{N-1} \widehat{U}_{ik} \circ U_{kj} \right)_{i,j\in[N]} \right. \\ \mathscr{V}_{\mathrm{Mult}} = \mathscr{V}_2 \diamond \mathscr{V}_1 = \left(\sum_{k=0}^{N-1} \widehat{V}_{ik} G^{-1}\left(V_{kj}\right) \right)_{i,j\in[N]} \\ = \left(A\left(\sum_{k=0}^{N-1} \widehat{U}_{ik} \circ U_{kj} \right) + \left(\sum_{k=0}^{N-1} \widehat{C}_{ik} C_{kj} \right) G \right)_{i,j\in[N]} \end{cases} \tag{13}$$

where, we define the operation $\circ$ as follows:

$$\widehat{U}_{ij} \circ U_{ij} = \widehat{C}_{ij} U_{ij} + \widehat{U}_{ij} G^{-1}\left(V_{ij}\right). \tag{14}$$

For simplicity, we take the first element of the above three matrices to illustrate the correctness of the multiplicative homomorphism:

$$\begin{aligned} & \widehat{V}_{00} G^{-1}(V_{00}) + \ldots + \widehat{V}_{N-10} G^{-1}(V_{0N-1}) \\ = \ & \left(A\widehat{U}_{00} + \widehat{C}_{00} G \right) G^{-1}(V_{00}) + \ldots \\ & + \left(A\widehat{U}_{N-10} + \widehat{C}_{N-10} G \right) G^{-1}(V_{0N-1}) \\ = \ & A\left(\left(\widehat{C}_{00} \right) U_{00} + \widehat{U}_{00} G^{-1}(V_{00}) + \ldots \right. \\ & \left. + \left(\widehat{C}_{0N-1} U_{N-10} + \widehat{U}_{0N-1} G^{-1}(V_{N-10}) \right) \right. \\ & + \left(\widehat{C}_{00} C_{00} + \ldots + \widehat{C}_{N-10} C_{0N-1} \right) G \\ = \ & A\left(\widehat{U}_{00} \circ U_{00} + \ldots + \widehat{U}_{N-10} \circ U_{0N-1} \right) \\ & + \left(\widehat{C}_{00} C_{00} + \ldots + \widehat{C}_{N-10} C_{0N-1} \right) G. \end{aligned} \tag{15}$$

Both A and G should be seen as a number while performing the multiplication operation described above.

Next, we do the analyzation to the corresponding noise boundary in the process of multiplicative homomorphism. From Equations (13) and (14), we have that:

$$\begin{aligned} < C_{\mathrm{Mult}}, \text{Powers of } 2(sk_r) > &= \mu_1\mu_2 \text{ Power of } 2(sk_r) + \mu_1 R_2 e_2 + C_2 R_1 e_1 \\ \left\| \mathscr{U}_{\mathrm{Mult.ij}} \right\|_\infty &= \left\| \sum_{k=0}^{N-1} \left(\widehat{U}_{ik} \circ U_{kj} \right)_{i,j\in[N]} \right\|_\infty \\ &\leq \left\| \sum_{k=0}^{N-1} \left(\widehat{C}_{ik} U_{kj} + \widehat{U}_{ik} G^{-1}\left(V_{kj}\right) \right)_{i,j\in[N]} \right\|_\infty \\ &\leq \sum_{k=0}^{N-1} \left(\left\| \widehat{C}_{ik} U_{kj} \right\|_\infty + \left\| \widehat{U}_{ik} G^{-1}\left(V_{kj}\right) \right\|_\infty \right). \end{aligned} \tag{16}$$

If R_1e_1 and R_2e_2 are bounded by α, then the upper noise boundary of C_{Mult} is $(N+1)\alpha$.

If U_1 and U_2 are bounded by β, then the upper noise boundary of $\mathscr{U}_{\mathrm{Mult}}$ is $N(m+1)\beta$.

Faster Homomorphic Multiplication. Given fresh signcryptions $(C_1, \mathscr{U}_1), \ldots, (C_S, \mathscr{U}_S)$, we can calculate that the upper noise boundary of $C_{MultS} = C_S C_{S-1} \ldots C_1 = (C_S(\ldots(C_3(C_2C_1))\ldots))$ is $(S-1)(N+1)\alpha$.

The upper noise boundary of $\mathscr{U}_{\mathrm{MultS}} = \mathscr{U}_S \otimes \mathscr{U}_{S-1} \otimes \ldots \otimes \mathscr{U}_1 = (\mathscr{U}_S \otimes (\ldots(\mathscr{U}_3 \otimes (\mathscr{U}_2 \otimes \mathscr{U}_1))\ldots))$ is $\frac{N^{S-1}-1}{N-1} N(m+1)\beta$.

Proof 7. From Equation (16), we have that:

$$\begin{aligned} \left\| e_{\mathrm{Mult2}} \right\|_\infty &= \left\| \mu_1 R_2 e_2 + C_2 R_1 e_1 \right\|_\infty = (N+1)\alpha, \\ \left\| \mathscr{U}_{\mathrm{Mult.ij}} \right\|_\infty &\leq \sum_{k=0}^{N-1} \left(\left\| \widehat{C}_{ik} U_{kj} \right\|_\infty + \left\| \widehat{U}_{ik} G^{-1}\left(V_{kj}\right) \right\|_\infty \right) \\ &= N(m+1)\beta. \end{aligned} \tag{17}$$

And we can get that:

$$\begin{aligned} \left\| e_{\mathrm{Mult3}} \right\|_\infty &= \left\| e_{\mathrm{Mult2}}(\mu_3 + R_3e_3) \right\|_\infty \\ &= \left\| \mu_3 e_{\mathrm{Mult2}} + R_3 e_3 e_{\mathrm{Mult2}} \right\|_\infty \\ &\leq \left\| \mu_3 e_{\mathrm{Mult2}} \right\|_\infty + \left\| R_3 e_3 e_{\mathrm{Mult2}} \right\|_\infty \\ &\leq 2(N+1)\alpha. \end{aligned} \tag{18}$$

According to recursion, we can obtain the corresponding noise boundary of signcryption $C_{\mathrm{MultS}} = C_S C_{S-1} \ldots C_1 = (C_S(\ldots(C_3(C_2C_1))\ldots))$ is $(S-1)(N+1)\alpha$.

Similarly, the upper noise boundary of $\mathscr{U}_{Mult} = (\mathscr{U}_S \otimes (\ldots(\mathscr{U}_3 \otimes (\mathscr{U}_2 \otimes \mathscr{U}_1))\ldots))$ is $\frac{N^{S-1}-1}{N-1} N(m+1)\beta$. □

Faster Homomorphic Verification. Due to the homomorphism of signcryption, we can verify the sum of all components in $\mathscr{U}$ to achieve public verification.

$$\mathsf{Verify}(\mathscr{C}, \mathscr{U}) = \mathsf{Verify}\left(\sum_{ij} C_{ij}, \sum_{ij} U_{ij}\right) \; i, j \in [N] \tag{19}$$

Completeness. In addition to the noise boundary analyzed above, we can also set appropriate parameters to ensure the correctness of homomorphic evaluation, thus ensuring the completeness of the proposed FHSC scheme. As shownin a study by Gentry et al. [5], we can evaluate a depth-S circuit of NANDs over α-bounded ciphertexts to obtain a $q/8$-bounded ciphertext if $q/\alpha > 8(N+1)^S$. The ciphertext can be correctly publicly verified if the noise boundary $\frac{N^{S-1}-1}{N-1} N(m+1)\beta < \beta_{\max}$. Therefore, the allowed evaluation depth S for our scheme is the minimum of above.

3.3. Security Analysis

Theorem 8 (IND-CPA security). *The proposed FHSC scheme satisfies IND-CPA security, if the GSW-FHE scheme satisfies IND-CPA security.*

Proof 9. We assume that there is a adversary $\mathscr{A}^*$ can break the IND-CPA security of FHSC scheme with a nonnegligible advantage in the security game. Then, the adversary $\mathscr{A}$ can break the IND-CPA security of the GSW-FHE scheme with a nonnegligible advantage utilizing the ability of $\mathscr{A}^*$. Actually, the advantage for $\mathscr{A}$ to break the IND-CPA security of the GSW-FHE scheme is negligible. Therefore, our FHSC scheme satisfies IND-CPA security.

In the game, the challenger $\mathscr{C}$ runs $\mathsf{GSW.Keygen}$ and generates a pair of key $(B, \mathbf{s})$ as GSW-FHE key and sends them to $\mathscr{A}$. Then, $\mathscr{A}$ chooses a pair of sender's key (A, td) and sends (B, A, td) to $\mathscr{A}^*$. Next, $\mathscr{A}^*$ chooses two messages μ_0 and μ_1, where $|\mu_0| = |\mu_1|$, and sends (μ_0, μ_1) to $\mathscr{A}$. Immediately, $\mathscr{A}$ sends μ_0, μ_1 to $\mathscr{C}$. Subsequently, $\mathscr{C}$ randomly selects $b \in \{0, 1\}$, generates C_b, and sends C_b to $\mathscr{A}$. Then $\mathscr{A}$ generates $\mathscr{U}_b$ for C_b and sends $(C_b, \mathscr{U}_b)$ to $\mathscr{A}^*$. The adversary $\mathscr{A}^*$ is supposed to return b' from the signcryption $(C_b, \mathscr{U}_b)$ and sends b' to $\mathscr{A}$. Both $\mathscr{A}$ and $\mathscr{A}^*$ win if $b' = b$. Under our assumption, $\mathscr{A}^*$ can break the IND-CPA security of FHSC scheme with a nonnegligible advantage. Furthermore, $\mathscr{A}$ can break the security of GSW-FHE scheme. Attributed to the security of the GSW-FHE scheme, the adversary $\mathscr{A}$ could not output $b' = b$ with a nonnegligible advantage. Therefore, we can get that our FHSC scheme satisfies IND-CPA security. □

Theorem 10 (Strong unforgeability). *If the forger $\mathscr{F}^*$ can break the strong unforgeability of FHSC scheme with a nonnegligible advantage, then the forger $\mathscr{F}$ can break the strong unforgeability of GVW-FHS scheme utilizing the ability of $\mathscr{F}^*$.*

Proof 11. Assuming that the forger $\mathscr{F}^*$ can break the FHSC scheme with a nonnegligible advantage in the security game with a nonnegligible advantage. Then, $\mathscr{F}$ can break the strong unforgeability of FHSC scheme with a nonnegligible advantage utilizing the ability of $\mathscr{F}^*$. Actually, the advantage for $\mathscr{F}$ to break the strong unforgeability of GVW-FHS scheme is negligible. Therefore, our FHSC scheme satisfies IND-CPA security.

In the game, the challenger $\mathscr{C}$ could run $\mathsf{GVW.Setup}$ to generate a pair of sender's key (A, td) as GVW-FHS key and sends A to $\mathscr{F}$. Then, $\mathscr{F}$ chooses a pair of receiver's key $(B, \mathbf{s})$ as GSW-FHE key and sends (A, B) to $\mathscr{F}^*$. Following that, $\mathscr{F}^*$ chooses plaintexts $(\mu_1, \mu_2, \ldots, \mu_S)$ and sends to $\mathscr{F}$. Given plaintexts, $\mathscr{F}$ generates corresponding ciphertexts $(C_1, \ldots, C_S)$ and sends them to $\mathscr{C}$ for signature queries. After signature queries, $\mathscr{F}$ could obtain $(\mathscr{U}_1, \ldots, \mathscr{U}_S)$ and sends $(C_1, \mathscr{U}_1), \ldots, (C_S, \mathscr{U}_S)$ to $\mathscr{F}^*$ as the result of signcryption queries for $\mathscr{F}^*$. Under our assumption, $\mathscr{F}^*$ can construct a valid signcryption $(C^*, \mathscr{U}^*)$ with a nonnegligible advantage, where $\mathscr{U}^* \neq f(\mathscr{U}_1, \ldots, \mathscr{U}_S)$ for $C^* = f(C_1, \ldots, C_S)$. Furthermore, $\mathscr{F}$ can construct a valid signature $\mathscr{U}^*$ with a nonnegligible advantage to break the strong unforgeability of GVW-FHS scheme. Attributed to the security of the GVW scheme, we can get that the forger cannot forge a new signature $\mathscr{U}^*$ with a nonnegligible advantage. Thereby, we can get that our FHSC scheme satisfies strong unforgeability. □

4. Conclusion and Open Problems

In this paper, we propose a leveled FHSC scheme with public verifiability and show its IND-CPA security and strong unforgeability under the standard assumption. Although we can utilize faster homomorphic verification to reduce the number of verifications, our scheme is still not practical enough. It's interesting to construct more efficient schemes.

Data Availability

Data availability is not applicable to this article as no new data were created or analyzed in this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

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

References

[1] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, pp. 169–178, 2009.

[2] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," in *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pp. 97–106, IEEE, Palm Springs, CA, USA, 2011.

[3] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical gapsvp," in *Advances in*

Cryptology–CRYPTO 2012: 32nd Annual Cryptology Conference, Proceedings, pp. 868–886, Springer, Santa Barbara, CA, USA, August 19–23, 2012.

[4] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," *ACM Transactions on Computation Theory (TOCT)*, vol. 6, no. 3, pp. 1–36, 2014.

[5] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," in *Advances in Cryptology–CRYPTO 2013: 33rd Annual Cryptology Conference*, Proceedings, Part I, pp. 75–92, Springer, Santa Barbara, CA, USA, August 18–22, 2013.

[6] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security*, Proceedings, Part I 23, pp. 409–437, Springer, Hong Kong, China, December 3–7, 2017.

[7] S. Gorbunov, V. Vaikuntanathan, and D. Wichs, "Leveled fully homomorphic signatures from standard lattices," in *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pp. 469–477, 2015.

[8] F. Wang, K. Wang, B. Li, and Y. Gao, "Leveled strongly-unforgeable identity-based fully homomorphic signatures," in *Information Security: 18th International Conference, ISC 2015*, Proceedings 18, pp. 42–60, Springer, Trondheim, Norway, September 9–11, 2015.

[9] R. Li, F. Wang, R. Zhang, and K. Chen, "NTRU-based fully homomorphic signature," *Security and Communication Networks*, vol. 2022, Article ID 9942717, 11 pages, 2022.

[10] Y. Zheng, "Digital signcryption or how to achieve cost (signature & encryption) ≪ cost (signature)+ cost (encryption)," in *Advances in CryptologyCRYPTO'97: 17th Annual International Cryptology Conference*, pp. 17–21, Springer, Santa Barbara, California, USA, 1997.

[11] F. Rezaeibagha, Y. Mu, S. Zhang, and X. Wang, "Provably secure homomorphic signcryption," in *Provable Security: 11th International Conference, ProvSec 2017*, pp. 349–360, Springer, Xi'an, China, October 23–25, 2017.

[12] S. Li, B. Liang, A. Mitrokotsa, and R. Xue, "Homomorphic signcryption with public plaintext-result checkability," *IET Information Security*, vol. 15, no. 5, pp. 333–350, 2021.

[13] X. Jin, F. Wang, R. Zhang, B. Lian, and K. Chen, "Leveled fully homomorphic signcryption from lattices," *IEEE Access*, vol. 11, pp. 35232–35242, 2023.

[14] R. Tsabary, "An equivalence between attribute-based signatures and homomorphic signatures, and new constructions for both," in *Theory of Cryptography: 15th International Conference, TCC 2017*, Proceedings, Part II, pp. 489–518, Springer, Baltimore, MD, USA, November 12–15, 2017.

[15] M. Ajtai, "Generating hard instances of lattice problems," in *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, pp. 99–108, 1996.

[16] D. Micciancio, "Almost perfect lattices, the covering radius problem, and applications to ajtai's connection factor," *SIAM Journal on Computing*, vol. 34, no. 1, pp. 118–169, 2004.

[17] D. Micciancio and O. Regev, "Worst-case to average-case reductions based on gaussian measures," *SIAM Journal on Computing*, vol. 37, no. 1, pp. 267–302, 2007.

[18] D. Micciancio and C. Peikert, "Hardness of sis and lwe with small parameters," in *Advances in Cryptology–CRYPTO 2013: 33rd Annual Cryptology Conference*, Proceedings, Part I, pp. 21–39, Springer, Santa Barbara, CA, USA, August 18–22, 2013.

[19] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *Journal of the ACM (JACM)*, vol. 56, no. 6, pp. 1–40, 2009.

[20] M. Ajtai, "Generating hard instances of the short basis problem," in *Automata, Languages and Programming: 26th International Colloquium, ICALP99 Prague*, Proceedings 26, pp. 1–9, Springer, Czech Republic, July 11–15, 1999.

[21] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pp. 197–206, 2008.

[22] J. B. Alwen and C. Peikert, "Generating shorter bases for hard random lattices," *Theory of Computing Systems*, vol. 48, no. 3, pp. 535–553, 2011.

[23] D. Micciancio and C. Peikert, "Trapdoors for lattices: simpler, tighter, faster, smaller," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 700–718, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.