IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 9, 2019, accepted January 20, 2019, date of publication February 26, 2019, date of current version May 9, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2900003*

# Attribute-Based Signcryption From Lattices in the Standard Model

**JIANHUA YAN[1,2], LICHENG WANG[2], MUZI LI[1], HASEEB AHMAD[3], JUN YUE[1], AND WENBIN YAO[2]**

[1]School of Information and Electrical Engineering, Ludong University, Yantai 264025, China
[2]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
[3]Department of Computer Science, National Textile University, Faisalabad 37610, Pakistan

Corresponding author: Jianhua Yan (asksky05@163.com)

This work was supported by National Key R&D Program of China (Grant No. 2016YFB0800602), Shandong Province Foundation (No. ZR2017MF035), Open Foundation of State key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications) (No. SKLNST-2016-2-19), Shandong Provincial Key R&D Program of China (Grant No. 2018CXGC0701), China 111 Project (Grant No. B08004), and Shandong Province Foundation (No. ZR2017MF062). This work is also supported by the Natural Science Foundation of Hebei Province (no. F2018408040), and the Hebei Education Funds for Youth Project (no. QN2018047).

**ABSTRACT** For realizing the fine-grained access control with non-interactive approach, and effectively guaranteeing the comprehensive security for information under the post-quantum environment, this paper proposes an attribute-based signcryption (ABSC) scheme based on the intractability of lattices. The proposed ABSC scheme is proved indistinguishable against the inner adaptive-chosen ciphertext attacks (IND-CCA2) and existentially unforgeable against inner chosen-message attacks (EUF-CMA), in the standard model. The theoretical analysis presents that the public key size and the computational cost of the signcryption operation are both reduced obviously, compared with the signature and then encryption mechanism. An efficient variant is also presented that significantly decreases the computational complexity of unsigncryption operation at the expense of an increase in the ciphertext size.

**INDEX TERMS** Signcryption, lattice, standard model, attribute-based, fine-grained access control.

## I. INTRODUCTION

As a distributed open computing environment, the Internet integrates computing resources and improves system utilization. Cloud computing reduces the hardware and software costs, energy consumption and system maintenance expense due to undertaking the computing and storage tasks of users. As a result, cloud computing has become a prospective form of computing. The user data is migrated from local devices to the cloud, which brings security issues to the data. In the multi-user access environment, it is the key issue of the correct utilization for data to reduce the storage cost and ensuring real-time consistency. Attribute-based cryptography has been proposed in time for non-interactive fine-grained access control [1]–[5].

The comprehensive security of the information is the valid approach to ensure data reliability in a high-level. A feasible method to ensure comprehensive security is to implement signature and encryption for message successively. However, the signcryption, proposed by Zheng [6], is a better alternative due to the much lower cost. The signcryption has important applications in many scenarios such as key management, electronic commerce, mobile communications, and smart cards. In 2011, signcryption was formally accepted as an international standard, ISO/IEC 29150:2011, by the International Organization of Standardization (ISO).

The high efficiency of signcryption and the flexibility of attribute-based cryptography make attribute-based signcryption (ABSC) very useful for handing out the secret information under the control of access policy. In fact, ABSC has many other advantages except realizing integrity, authentication, non-repudiation, confidentiality and fine-grained access with lower cost. ABS enjoys perfect privacy, unlinkability and collusion resistant unforgeability. So, it is widely applied in anonymous authentication, trust negotiations [7] and attribute-based messaging. When ABS is integrated into ABSC, the anonymity of the signature in ABSC is further strengthened. The anonymity of ABE is achieved by the policy. The ABE is divided into ciphertext policy attribute-based encryption (CP-ABE) and key policy attribute-based

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

encryption (KP-ABE), according to the difference that the policy is imposed upon ciphertext and private key. Indeed, ABSC recently has been extensively studied and many fruits [1]–[4], [8]–[10] have been obtained.

The existing ABSC schemes [1]–[3], [4] are all based on the intractability assumptions from the number theory. The blossom of quantum computation badly menaces their hardness assumptions [11], [12]. As a result, it urgently needs to design an ABSC scheme with ability to resist the quantum attacks. Lattice is extensively believed to be one of the most powerful cryptography tools to stand up the quantum attacks. In addition, lattice-based cryptography has many other attractive properties. It has high asymptotic computational efficiency due to only involving moderate modular additions and moderate modular multiplications. It is very flexible to be used to realize some complex cryptographic primitives and tools such as fully homomorphic encryption [13]–[17], fully homomorphic signature [18], [19] and multilinear map [20], [21]. Xiang *et al.* [8] made a progress to construct an ABSC scheme based on lattice by conferring private keys according to attributes. However, the ciphertexts are generated also according to attributes. As a result, this scheme [8] cannot express the access policy well and does not enjoy the flexibility of the policy. Therefore, it has an important theoretical and practical significance to construct a true ABSC scheme based on lattice.

### A. OUR CONTRIBUTION

In this paper, we propose an ABSC from lattices. Our contributions are summarize as follows:

- To realize the fine-grained access control in a non-interactive way and resist the known quantum attacks, we construct a key policy attribute-based signcryption scheme based on the lattice hardness assumptions by borrowing the private key extraction technique for policy from [22]. Meanwhile, in the proposed ABSC, an attribute based signature scheme from lattices is constructed. The anonymity of the signer is improved. Only the group satisfying a particular policy can learn the attribute information of the signer. However, in the signature and then encryption (StE) with identical security, the attributes of signer are leaked to the public.
- For hiding the signature value and shortening the ciphertext size, the public encryption section used in our scheme is a variant of Regev encryption [23], denoted by $\mathcal{E}_G$. This variant $\mathcal{E}_G$ is not semantically secure. It was made IND-CCA secure with FO technique, such as [24], but the hash function is replaced with a signature. However, it is only proved secure in the random oracle. In fact, the encryption $\mathcal{E}_G$ is always assembled with another part of ciphertext to form an IND-CPA secure scheme, such as [23], [25], and [26]. Different from the above schemes, we combine it with a new section of ciphertext with small size to get an IND-CCA1 secure scheme in the standard model. That is, the security reduction is achieved by using the trapdoor switching technique based on the leftover hash lemma. The simple and efficient Exclusive-Or operation is employed to hide the original information. Thus, the proposed scheme is efficient to guarantee the security of data especially with big size. The proposed ABS is deliberately increased a segment with a small size, such that it can be proved unforgeable against the inner selective attribute set adaptive-chosen message attacks.
- The encryption section and the signature section are closely tangled together to strengthen the security. Not only the non-malleability of encryption is guaranteed by the preceding signature, but also the tag used for encryption is selected according to the preceding signature rather than by encryptor. In a word, the IND-CCA1 secure scheme is enhanced to IND-CCA2 security by reusing the function of the signature for the message. However, it leads to difficulty in security reduction. To vanish the trapdoor in reduction, we utilize a chameleon hash function. In the proposed scheme, the computational overhead of signcryption decreases more than 50%, compared to StE. And the unsigncryption cost of the variety scheme reduces about 50%.

### B. PAPER OUTLINE

This paper is organized as follows. The necessary preliminaries are introduced in Section 2. The primitive and the security models of ABSC are reviewed in Section 3. In Section 4, the proposed scheme is presented in detail, followed by the consistency proof and the security reduction. The performance analysis and a variant are given in Section 5. Finally, the concluding remarks are drawn in Section 6.

## II. PRELIMINARIES

In this paper, the notions and the corresponding meaning are as follows. $\mathbb{Z}/\mathbb{R}$ : the set of integers / real numbers; $\mathbb{T}$: real interval $[0, 1)$; $\mathbb{Z}_q$: residue class mod $q$. $\mathbb{Z}^n/\mathbb{Z}_q^n$, $\mathbb{R}^n$ : vectors space on $\mathbb{Z}/\mathbb{Z}_q$, $\mathbb{R}$; $\mathbb{Z}^{n\times m}/\mathbb{Z}_q^{n\times m}$: matrices space on $\mathbb{Z}/\mathbb{Z}_q$. lower-case and bold letters: vectors; upper-case and bold letters: matrices. $\tilde{\mathbf{A}}$: Gram-Schmidt orthogonalization of $\mathbf{A}$; $s_1(\cdot)$: the largest singular value of a matrix; $\|\cdot\|$ the maximum norm of the column vectors in a matrix or the norm of a vector. $\mathbf{s} \xleftarrow{\$} U(P)$: uniformly choose from $P$; $\mathbf{s} \leftarrow \chi(\mathcal{P})$ : choose from $P$ according to the distribution $\chi$; $\mathbf{s} \in \chi/\mathbf{s} \xleftarrow{\$} \chi$ for short. $[k]$: $\{1, 2, \cdots, k\}$. $|\cdot|$: the length of bit string; $\|$: horizontally concatenate matrices or vectors.

### A. LATTICE AND GAUSSIAN DISTRIBUTION

*Definition 1 (Lattice): Lattice is a discrete additive subgroup of* $\mathbb{R}^m$ : $\Lambda = \mathcal{L}(\mathbf{B}) = \{\mathbf{Bx} = \Sigma_{i=1}^n x_i\mathbf{b_i}|\mathbf{x} \in \mathbb{Z}^n\}$, *where the linearly independent vectors* $\mathbf{B} = \{\mathbf{b_1}, \mathbf{b_2}, \cdots, \mathbf{b_n}\}$ *constitute a basis. In fact, the q-ary lattice* $\Lambda^{\perp}(\mathbf{A}) = \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{Az} = \mathbf{0} \bmod q\}$ *is more frequently used.*

The security of the proposed scheme depends on LWE and SIS problems, whose definitions and intractability are as follows.

For integers $n > 0$ and $q > 2$, $\mathbf{A}_{\mathbf{s},\chi}$ indicates the distribution of $(\mathbf{a}, \mathbf{a}^t\mathbf{s} + x)$ over $\mathbb{Z}_q^n \times \mathbb{Z}_q$, where $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$ and $x$ is chosen from some distribution $\chi$.

*Definition 2 (Learning With Errors (LWE) [23]): Given an integer $q = q(n)$ and a distribution $\chi$ over $\mathbb{Z}_q$, the learning with errors problem $LWE_{q,\chi}$ is to distinguish $\mathbf{A}_{\mathbf{s},\chi}$ from $U(\mathbb{Z}_q^n) \times U(\mathbb{Z}_q)$ with non-negligible probability, where $\mathbf{s} \xleftarrow{\$} U(\mathbb{Z}_q^n)$.*

For $\alpha \in \mathbb{R}^+$, $\Psi_\alpha$ denotes the distribution of a normal variable with mean 0 and standard deviation $\alpha/\sqrt{2\pi}$, reduced modulo 1. For $x \leftarrow \Psi_\alpha$, $\bar{\Psi}_\alpha$ is the discretized normal distribution on $\mathbb{Z}_q$, namely $\lfloor q \cdot x \rceil \bmod q$, where $\lfloor \rceil$ represents rounding.

*Proposition 1 (Hardness of LWE [23], [27]): For $\alpha = \alpha(n) \in (0, 1)$ and a prime $q = q(n)$ satisfying $\alpha q > 2\sqrt{n}$, the $LWE_{q,\bar{\Psi}_\alpha}$ is as hard as approximating $SIVP_\gamma$ within $\tilde{O}(n/\alpha)$ factors (referring to [28] for its hardness) in the worst case.*

*Definition 3 (Small Integer Solution (SIS) [29]): Given an integer $q$, a real $\beta > 0$ and a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, the aim of $SIS_{q,\beta}$ is to find $\mathbf{0} \neq \mathbf{z} \in \mathbb{Z}^m$ such that $\mathbf{Az} = \mathbf{0} \bmod q$ and $\|\mathbf{z}\| \leq \beta$.*

*Proposition 2 (Hardness of SIS Theorem 5.16 [29]): For poly-bounded $m$, $\beta = poly(n)$ and prime $q \geq \beta \cdot \omega(\sqrt{n \log n})$, if there is an efficient algorithm to solve $SIS_{q,\beta}$ in average case, then there exists an efficient algorithm to solve the approximating $SIVP\gamma$ problem in the worst case, where $\gamma = \beta \cdot \tilde{O}(\sqrt{n})$.*

In lattice-based cryptography, an especial distribution, namely discrete Gaussian distribution, is frequently used, which has the following beautiful properties.

*Definition 4 (Discrete Gaussian distribution [29]): For a vector $\mathbf{c}$, real $s > 0$, and lattice $\Lambda$, the discrete Gaussian distribution over $\Lambda$ is defined as $D_{\Lambda,s,\mathbf{c}}(\mathbf{x}) = \frac{D_{s,\mathbf{c}}(\mathbf{x})}{D_{s,\mathbf{c}}(\Lambda)} = \frac{\rho_{s,\mathbf{c}}(\mathbf{x})}{\rho_{s,\mathbf{c}}(\Lambda)}$, $\forall \mathbf{x} \in \Lambda$, where $\rho_{s,\mathbf{c}}(\mathbf{x}) = e^{-\pi\|(\mathbf{x}-\mathbf{c})/s\|^2}$.*

*Proposition 3: Let $m > 2n \log q$, $\mathbf{B}$ be a basis of $\Lambda^\perp(\mathbf{A})$ for $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, $s \geq \|\tilde{\mathbf{B}}\|\omega(\sqrt{\log n})$. The discrete Gauss distribution has the following properties.*

1) *(Theorem 3.1 [25] ) When $\mathbf{x} \leftarrow D_{\mathbb{Z}^m,s}$, the distribution of $\mathbf{y} = \mathbf{Ax} \in \mathbb{Z}_q^n$ is negl(n)-far from the $U(\mathbb{Z}_q^n)$. Given $\mathbf{y}$, the conditional distribution of $\mathbf{x}$ is statistically close to $D_{\Lambda_{\mathbf{y}}^\perp(A),s}$.*
2) *(Lemma 4.4 [29]) $Pr_{\mathbf{x}\sim D_{\Lambda,s,\mathbf{v}}}\{\|\mathbf{x} - \mathbf{v}\| > s\sqrt{n}\} \leq \frac{1+\epsilon}{1-\epsilon} \cdot 2^{-n}$.*
3) *( [25] ) For arbitrary $\mathbf{y} \in \mathbb{Z}_q^n$ the min-entropy of $\mathbf{y}$'s pre-image $\mathbf{x}$, namely $\mathbf{Ax} = \mathbf{y}$ and $\|x\| \leq s\sqrt{m}$, is at least $\omega(\log n)$.*

In the scheme design and security reduction, a special matrix $\mathbf{R} \xleftarrow{\$} \{-1, 1\}^{k\times m}$ is used, because it has small norm and satisfies leftover hash lemma.

*Proposition 4 (Lemma 15 [26]): For $\mathbf{R} \xleftarrow{\$} \{-1, 1\}^{k\times m}$, there exists a universal constant $C$ such that $Pr[\|\mathbf{R}\| > C\sqrt{k+m}] < e^{-(k+m)}$. In fact, It is sufficient to set $C = 12$.*

*Proposition 5 (Leftover Hash Lemma, Lemma 13 [26]): The matrices $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{n\times k}$, $\mathbf{R} \xleftarrow{\$} \{1, -1\}^{m\times k}$, where $q > 2$ is a prime, $m > (n+1)\log_2 q + \omega(\log n)$ and $k = k(n)$ is the polynomial size of n. Then, the distribution of $(\mathbf{A}, \mathbf{AR}, \mathbf{R}^t\mathbf{z})$ is within negligible statistical distance from the distribution $(\mathbf{A}, \mathbf{B}, \mathbf{R}^t\mathbf{z})$ for arbitrary $\mathbf{z} \in \mathbb{Z}_q^m$.*

### B. RELATED ALGORITHMS

In the scheme design, some underlying algorithms are used, such as trapdoor generation algorithm, short lattice vector sampling algorithm and basis sampling algorithm.

*Proposition 6 (Trapdoor Generation Algorithm, Theorem 3.2 [30]):* For some fixed real $\delta > 0$, integer $q > 2$ and integer $m \geq (5 + 3\delta)n \log q$, the algorithm $\mathbf{TrapGen}(n, q)$ outputs $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{S} \in \mathbb{Z}_q^{m\times m}$ in polynomial time, such that $\mathbf{A}$ is within negligible statistical distance from $U(\mathbb{Z}_q^{n\times m})$, $\|\mathbf{S}\| \leq O(n \log q)$ and $\|\tilde{\mathbf{S}}\| \leq O(\sqrt{n \log q})$, with overwhelming probability.

*Proposition 7 (Left Sampling Algorithm, Theorem* 17 *[26]): Let integers $q > 2, m > n$. The algorithm* $\mathbf{SampleLeft}(\mathbf{A}, \mathbf{B}, \mathbf{T_A}, \mathbf{y}, s)$ *outputs $\mathbf{x} \in \mathbb{Z}^{m+m'}$ within negligible statistical distance with $D_{\Lambda_q^{\mathbf{y}}(\mathbf{A}\|\mathbf{B}),s}$, where $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{B} \in \mathbb{Z}_q^{n\times m'}$, $s > \|\widetilde{\mathbf{T_A}}\|\omega(\sqrt{\log(m+m')})$, $\mathbf{y} \in \mathbb{Z}^m$ and $\mathbf{T_A}$ is a trapdoor for $\Lambda^\perp(\mathbf{A})$.*

*Proposition 8 (Right Sampling Algorithm, Theorem* 19 *[26]):* Let integers $q > 2, m > n$. Input $\mathbf{A} \in \mathbb{Z}_q^{n\times m'}$, $\mathbf{R} \in \mathbb{Z}_q^{m'\times m}$, $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$, invertible matrix $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$, the trapdoor $\mathbf{T_B}$ of $\Lambda^\perp(\mathbf{B})$, $s > \|\widetilde{\mathbf{T_B}}\| \cdot s_R\omega(\sqrt{\log m})$ and $\mathbf{y} \in \mathbb{Z}^m$, the algorithm $\mathbf{SampleRight}(\mathbf{A}, \mathbf{B}, \mathbf{H}, \mathbf{R}, \mathbf{T_B}, \mathbf{y}, s)$ outputs $\mathbf{x} \in \mathbb{Z}^{m+m'}$ within negligible statistical distance with $D_{\Lambda_q^{\mathbf{y}}(\mathbf{F},s)}$, where $\mathbf{F} = [\mathbf{A}\|\mathbf{AR} + \mathbf{HB}]$.

*Proposition 9 (Basis Sampling, Lemma* 29*, Corollary* 30*, Corollary* 31 *[26]):* 1) Input $\mathbf{A} \in \mathbb{Z}_q^{n\times m}, \mathbf{B} \in \mathbb{Z}_q^{n\times m'}$, invertible $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$, $\mathbf{R} \in \mathbb{Z}_q^{m\times m}$, a trapdoor $\mathbf{T_B}$ of $\Lambda^\perp(\mathbf{B})$ and $s > \|\widetilde{\mathbf{T_B}}\| \cdot s_R\omega(\sqrt{\log m})$, the algorithm $\mathbf{SampleBasisRight}(\mathbf{A}, \mathbf{B}, \mathbf{H}, \mathbf{R}, \mathbf{T_B}, s)$ runs $\mathbf{SampleRight}$ less than $O(m \log m)$, w.o.p $2m$, times with $\mathbf{y} = \mathbf{0}$, then outputs a basis $\mathbf{T}$ for $\Lambda^\perp(\mathbf{F})$, where $\mathbf{F} = [\mathbf{A}\|\mathbf{AR} + \mathbf{HB}]$, $\|\mathbf{T}\| \leq s\sqrt{m}$.

2) Input $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{C} \in \mathbb{Z}_q^{n\times m'}$, $s > \|\widetilde{\mathbf{T_A}}\|\omega(\sqrt{\log(m+m')})$ and a trapdoor $\mathbf{T_A}$ for $\Lambda^\perp(\mathbf{A})$, the algorithm $\mathbf{SampleBasisLeft}(\mathbf{A}, \mathbf{B}, \mathbf{T_A}, s)$ runs $\mathbf{SampleLeft}$ to output a basis $\mathbf{T}'$ for $\Lambda^\perp(\mathbf{F}')$ where $\mathbf{F}' = [\mathbf{A}\|\mathbf{C}]$ and $\|\mathbf{T}'\| \leq s\sqrt{m}$.

3) Especially, for identical $\mathbf{A}$, $s$ and $\mathbf{C} = \mathbf{AR}+\mathbf{HB}$, the two bases $\mathbf{T}$ and $\mathbf{T}'$ are statistically close.

*Proposition 10 (Pre-image Sampling Algorithm, [25]): Let integers $q > 2, m > n$.* $\mathbf{SamplePre}(\mathbf{A}, \mathbf{T_A}, \mathbf{y}, s)$ *takes inputs $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, the trapdoor $\mathbf{T_A}$ for $\Lambda^\perp(\mathbf{A})$, $\mathbf{y} \in \mathbb{Z}^m$ and $s > \|\widetilde{\mathbf{T_A}}\|\omega(\sqrt{\log(m)})$, outputs a pre-image $\mathbf{x} \in \mathbb{Z}^m$ within negligible statistic distance from $\mathcal{D}_{\Lambda_q^{\mathbf{y}}(\mathbf{A}),s}$ and $\mathbf{Ax} = \mathbf{y}$.*

In the Pre-image algorithm, a solution vector $\mathbf{x}' \in \mathbb{Z}_q^n$ is computed by solving equation $\mathbf{Ax}' = \mathbf{y} \bmod q$. Next, a vector $\mathbf{z}$ is selected randomly under the condition that $\mathbf{z}$

belongs to $\Lambda^{\perp}(\mathbf{A})$ and $\mathbf{z}$ is close to $-\mathbf{x}'$. Then, the vector $\mathbf{x} = \mathbf{z} - (-\mathbf{x}')$ is output. $\mathbf{Ax} = \mathbf{A}[\mathbf{z} - (-\mathbf{x}')] = \mathbf{Az} + \mathbf{Ax}' = \mathbf{Ax}' = \mathbf{y} \mod q$.

### C. UNIVERSAL HASH FUNCTION AND CHAMELEON HASH FUNCTION

*Definition 5 (Universal Hash Functions [31]): A family of hash functions $\mathcal{H} = \{h : \mathcal{X} \to \mathcal{Y}\}$ is called universal if for every distinct pair $x, x' \in \mathcal{X}$, $Pr_{h \leftarrow \mathcal{H}}[h(x) = h(x')] = 1/|\mathcal{Y}|$ holds.*

We introduce the conception about chameleon hash function by the example of the construction based on lattice.

*Proposition 11 (Lemma 4.1 of [32]): Let $n \geq 1, q \geq 2$, $m = O(n \log q)$, $k \geq 1$ be integers and $s = O(\sqrt{n \log q})$ be real. With respect to $\mathbf{A_0} \in \mathbb{Z}_q^{n \times k}$, $\mathbf{A_1} \in \mathbb{Z}_q^{n \times m}$, define hash function $h_{\mathbf{A}} : \mathcal{M} \times \mathcal{R} \to \mathcal{Y}$ as $h_{\mathbf{A}}(\mathbf{m}, \mathbf{r}) = \mathbf{A}(\mathbf{m} \| \mathbf{r}) = \mathbf{A_0 m} + \mathbf{A_1 r}$, where $\mathcal{M} \in \{0,1\}^k$, $\mathcal{R} = \{\mathbf{r} \leftarrow \mathcal{D}_{\mathbb{Z}^m, s}\}$, $\mathbf{A} = [\mathbf{A_0} \| \mathbf{A_1}]$ and $\mathcal{Y} = \mathbb{Z}_q^n$ are message space, randomness space and range, respectively. When $\Lambda^{\perp}(\mathbf{A_1})$ has a trapdoor, the hash family $\mathcal{H} = \{h_{\mathbf{A}}\}$ is a chameleon hash function family, supposing the hardness of $SIS_{q,\beta}$ for $\beta = \sqrt{k + 4\, s^2 m}$.*

The chameleon hash functions have the following 4 properties: efficient forward computation, collision-resistance, uniformity and chameleon property. The chameleon property is that the hash functions have ability to find a collision for any given input by utilizing the known trapdoor.

## III. ATTTRIBUTE-BASED SIGNCRYPTION: PRIMITIVE AND SECURITY MODELS

*Definition 6 (Attribute-Based Signcryption): AnAttribute-Based signcryption scheme consists of the following four algorithms:*

- **Setup**$(1^{\varpi})$*: The private key generator (PKG) executes the setup algorithm to initialize the system. It takes inputs a security parameter $1^{\varpi}$, publishes public parameters $\mathcal{P}_p$, master public key MPk and keeps the master secret key MSk as secret.*
- **Extract**$(MPk, MSk, Policy)$*: In this algorithm, PKG takes inputs the master public key MPk, the master secret key MSk, and an access policy Policy, and returns a corresponding private key $SK_{Plc}$ for Policy.*
- **Signcrypt**$(\mu, MPk, Policy_s, SK_{Plcs}, As_s, Policy_r)$*: In this algorithm, the sender takes inputs a message $\mu$, the master public key MPk, the policy $Policy_s$ satisfied by the senders, the secret key $SK_{Plcs}$ corresponding to $Policy_s$, the sender's attribute set $As_s$, and the policy $Policy_r$ satisfied by receivers, then generates a corresponding ciphertext c for $\mu$.*
- **Unsigncrypt**$(c, MPk, As_s, Policy_r, SK_{Plcr}, As_r)$*: In this algorithm, the receiver uses the information of the master public key MPk, the policy $Policy_r$ satisfied by the receivers, the secret key $SK_{Plcr}$ corresponding to $Policy_r$, the receiver's attribute set $As_r$, and the sender's attribute set $As_s$ to decrypt the ciphertext c and outputs the corresponding plaintext $\mu$.*

*Definition 7 (Consistency of Signcryption):* Define the successful probability of the unsigncryption for a signcryption scheme as follows.

$$p = \Pr\left[\begin{array}{l} \mathcal{P}_p \leftarrow \mathbf{Setup}(1^{\varpi}); \\ SK_{Plcr} \leftarrow \mathbf{Extract}(MPk, MSk, Policy_r); \\ SK_{Plcs} \leftarrow \mathbf{Extract}(MPk, MSk, Policy_s); \\ c \leftarrow \mathbf{Signcrypt}(\mu, MPk, As_s, Policy_s, SK_{Plcs}, Policy_r); \\ \mu' \leftarrow \mathbf{Unsigncrypt}(c, MPk, As_s, As_r, Policy_r, SK_{Plcr}) : \mu' = \mu \end{array}\right]$$

The signcryption scheme is called consistent, if and only if $1 - p$ is negligible.

For defining the confidentiality of an ABSC scheme, we give **Game IND-sAtt-CCA2** between the challenger and a probabilistic polynomial time (PPT) adversary as follows, by referring to [1], [8].

**Game IND-sAtt-CCA2**

- Initial: $\mathcal{A}$ announces the challenge attribute set $As_r^*$. $\mathcal{C}$ executes **Setup**$(1^{\varpi})$ to generate and publish $\mathcal{P}_p$ and $MPk$ to $\mathcal{A}$, but keeps $MSk$ to itself.
- Phase 1: $\mathcal{A}$ implements polynomially bounded queries:
  - **Extract**$(Policy)$: $\mathcal{A}$ submits a policy $Policy$ to $\mathcal{C}$ for private key query. If $As_r^*$ satisfies $Policy$, $\mathcal{C}$ replies $\perp$. Otherwise, $\mathcal{C}$ answers with the corresponding private key for $Policy$.
  - **Signcrypt**$(\mu, As_s, Policy_s, Policy_r)$: $\mathcal{A}$ sends a message $\mu$, the sender's attribute set $As_s$, the policy $Policy_s$ and the receiver's policy $Policy_r$ for signcryption query. If $As_s$ meets $Policy_s$, $\mathcal{C}$ returns the corresponding ciphertext. Otherwise $\mathcal{C}$ returns $\perp$.
  - **Unsigncrypt**$(c, As_s, Policy_r)$: $\mathcal{A}$ submits a ciphertext $c$, the sender's attribute set $As_s$ and the policy $Policy_r$ satisfied by the receiver's attribute set to $\mathcal{C}$ for unsigncryption queries. If $c$ is a valid ciphertext, $\mathcal{C}$ returns the corresponding plaintext; otherwise, $\mathcal{C}$ returns $\perp$.
- Challenge: $\mathcal{A}$ sends two isometric message plaintexts $\mu_0, \mu_1$, the sender's attribute set $As_s$, the policy $Policy_s$ satisfied by the senders and the policy satisfied by the receivers $Policy_r$ to $\mathcal{C}$. $\mathcal{C}$ randomly selects a bit $b \in \{0,1\}$. $\mathcal{C}$ executes **Signcrypt** for $u_b$ to get $c^*$, and sends $c^*$ to $\mathcal{A}$.
- Phase 2: $\mathcal{A}$ repeats the queries as in phase 1, except the unsigncryption query on $c^*$. The probability that the ciphertext generated normally under the other attribute sets equals $c^*$ is less than $q^{-[(|Att_r|+1)m+1]}$, and it is negligible.
- Guess: $\mathcal{A}$ outputs its guess $b' \in \{0,1\}$ for $b$.

The advantage of $\mathcal{A}$ to win **Game IND-sAtt-CCA2** is defined as $Adv(\mathcal{A}) = |\Pr[b = b'] - \frac{1}{2}|$.

*Definition 8 (Confidentiality of Signcryption): If there is no PPT adversary who can win **Game IND-sAtt-CCA2** with non-negligible advantage, then the corresponding attribute-based signcryption scheme is called indistinguishable against inner selective attribute adaptive-chosen ciphertext attacks (IND-sAtt-CCA2).*

To capture the unforgeability, we introduce **Game EUF-sAtt-CMA** played between the challenger $\mathcal{C}$ and a PPT forgery $\mathcal{F}$.

**Game EUF-sAtt-CMA**

- Initial: This step is totally identical to that in **Game EUF-sAtt-CMA**.
- Query: $\mathcal{A}$ executes polynomially bounded Extract, Signcrypt and Unsigncrypt queries as in **Game IND-sAtt-CMA**.
- Forgery: $\mathcal{F}$ outputs a tuple $(\mu, c, Policy_r, As_s^*)$, where $c$ is a valid ciphertext for $\mu$ under the sender's attribute set $As_s^*$, and the policy $Policy_r$ satisfied by the receivers.

Define the advantage of $\mathcal{F}$ to win **Game EUF-sAtt-CMA** as

$$Adv(\mathcal{F}) = \Pr[(\mu, \sigma) = \mathbf{Unsigncrypt}(c^*, As_s^*, Policy_r) \wedge NS],$$

where $NS$ is the Boolean value for the fact that $\sigma$ is a new signature for $\mu$.

*Definition 9 (Existential Unforgeability of Signcryption): An attribute-Based signcryption scheme is called existentially unforgeable against inner selective attribute adaptive-chosen message attacks (EUF-sAtt-CMA), if there exists no PPT inner forger who can win **Game SUF-sAtt-CMA** with non-negligible advantage.*

## IV. ATTRIBUTE-BASED SIGNCRYPTION SCHEME FROM LATTICES

### A. ENCODING FOR POLICY

Lewko and Waters [33] proposed a linear time algorithm (LW algorithm for short) to translate a Boolean circuit corresponding to access policy into a Linear Span Program (LSP) matrix. The core idea is as follows. The Boolean circuit can be expressed by a binary tree. Every node in the tree corresponds to a binary string, which will be discretized into a row of the LSP matrix. The binary string of a parent node is the computing result of the strings of its sons under the operator corresponding to this node. Appoint the string of the root node as some string. Then, the strings of nodes can be computed from root to leaves.

The concrete procedure of the encoding is as follows. (1) Express the Boolean circuit with a binary tree, in which the internal nodes are operators and the leaf nodes are attributes. Meanwhile record the number $\ell$ of AND gates. (2) Let $c_A = 1$, where $c_A$ is a counter for AND gate. Label the root node with $1|0^{\ell}$. (3) Encode for every node from root to leaves. Suppose the current node labeled with $s|0^{c_A}$, where $s$ is a bit string with length $\ell + 1 - c_A$. If this node corresponds to AND gate, label its children with $s|1|0^{c_A-1}$ and $0^{\ell-c_A}|-1|0^{c_A-1}$, respectively, since $s|1|0^{c_A-1}+0^{\ell-c_A}| - 1|0^{c_A-1} = s|0^{c_A}$. Meanwhile, set $c_A = c_A + 1$. If this node corresponds to OR gate, label its children both with $s|0^{c_A}$. (4)Discretize the label for every leaf node to a vector. Then, construct an LSP matrix by using all the vectors.

### B. CONSTRUCTION

- Setup$(1^{\varpi}, k)$: Take inputs a security parameter $1^{\varpi}$ and the maximum of attributes $k$:
  1) Select suitable integers $\iota, \varsigma$.
  2) Choose hash functions:
     - $H_0 : \{0,1\}^* \to \{0,1\}^{\iota'}$;
     - $H_1 : \{0,1\}^* \to \{0,1\}^{\varsigma}$;
     - $H_2 : \{0,1\}^* \to \{0,1\}^*$;
     - $H_3 : \{0,1\}^* \to \mathbb{Z}_q^m$;
     - $H_N : \{0,1\}^{\iota} \times \{\mathbf{r} \in D_{\mathbb{Z}^m,s}\} \to \mathbb{Z}_q^n$, where $s = O(\sqrt{n \log q}) \cdot \omega(\sqrt{\log m})$. The form of $H_N$ is similar with that of the chameleon hash functions in [32]. Specifically, $H_N$ is specified by matrix $\mathbf{N} = [\mathbf{N_0} \| \mathbf{N_1}]$, where $\mathbf{N_0} \in \mathbb{Z}_q^{n \times \varsigma}$ and $\mathbf{N_1} \in \mathbb{Z}_q^{n \times m}$.
  3) Choose the full-rank differences (FRD) encoding [26] $H : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n \times n}$.
  4) Choose suitable Gaussian parameters $s_1, s_2$.
  5) Generate public key and private key:
     - $(\mathbf{T_i}, \mathbf{A_i}) \leftarrow \mathbf{KeyGen}(1^{\varpi})$ for $i \in [k]$.
     - Choose $\mathbf{A_0}, \mathbf{B}, \mathbf{C}, \mathbf{D_i} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$ for $i \in [\iota']$.
     - Choose $v \xleftarrow{\$} \mathbb{Z}_q^n$.
  6) Output the master public key and keep the master private key as secret:
     $MPk = (\{\mathbf{A_i}\}_{i \in [k]}, \mathbf{A_0}, \mathbf{B}, \mathbf{C}, v, H_0, H_1, H_2, H_3, H_N)$, $MSk = (\{\mathbf{T_i}\}_{i \in [k]})$.
- Extract$(Pub, MSk, Plc)$: Take inputs the public key $Pub$, master secret key $MSk$ and Policy $Plc$, generate a secret key $SK_{Plc}$ for the policy $Plc$:
  1) Translate the policy $Plc$ into a LSP matrix $\mathbf{L}$ with LW algorithm. Suppose $\mathbf{L} \in \mathbb{Z}^{k \times \lambda}$, without loss of generality..
  2) Choose a temporary matrices $\mathbf{Z_i} \xleftarrow{\$} \mathbb{Z}^{n \times m}$ for $1 \le i \le \lambda$.
  3) Construct matrices,
     $\mathbf{M}_l = diag(\mathbf{A_1}, \mathbf{A_2}, \cdots, \mathbf{A_k})$,
     $\mathbf{M}_r = [\mathbf{L}_{(0)} \otimes \mathbf{A_0} \| \mathbf{L}_{(1)} \otimes \mathbf{Z_1} \| \cdots \| \mathbf{L}_{(\lambda)} \otimes \mathbf{Z}_{\lambda}]$,
     $\mathbf{M} = [\mathbf{M}_l \| \mathbf{M}_r]$, where $\mathbf{L}_{(i)}$ denotes the $i$-th column of $\mathbf{L}$.
  4) Obtain a short basis for $\Lambda^{\perp}(\mathbf{M})$ by calling $\mathbf{K^{(1)}} = \mathbf{SampleBasisLeft}(\mathbf{M}_l, \mathbf{M}_r, \mathbf{K^{(2)}}, s_2)$, where $\mathbf{K^{(2)}} = diag(\mathbf{T_1}, \mathbf{T_2}, \cdots, \mathbf{T_k})$. Note that $\mathbf{K^{(2)}}$ is a short basis for $\Lambda^{\perp}(\mathbf{M}_l)$ because $\mathbf{B_i}$ is a short basis of $\Lambda^{\perp}(\mathbf{A_i})$ for $i \in [k]$.
  5) Let $\mathbf{K} \in \mathbb{Z}^{(k+1)m \times (k+1)m}$ be the upper left sub-matrix of $\mathbf{K^{(1)}}$.
  6) Give $(\mathbf{K}, \mathbf{L})$ to the users satisfying $Plc$ as the secret key for the policy $Plc$.
- Signcrypt$(\mu, \mathbf{L_{ps}}, SK_{L_{ps}}, Att_s, Plc_r)$: On input a message $\mu$, the sender's attribute set $Att_s$, the policy matrix $\mathbf{L_{ps}}$ satisfied by $Att_s$, the private key $SK_{L_{ps}}$ corresponding to $\mathbf{L_{ps}}$ and the policy $Plc_r$ satisfied by the receivers, the sender does:

1) Compute $\upsilon = H_0(r_1, \mu, Att_s, \mathbf{L_{ps}}, Plc_r)$, where $r_1 \xleftarrow{\$} \{0,1\}^{\iota}$.
2) Compute $\mathbf{D}_\upsilon = \sum_{i=0}^{\iota'}(-1)^{\upsilon[i]}\mathbf{D_i}$, where $\upsilon[i]$ denotes the $i$-th bit of $\upsilon$.
3) Sample $\sigma_\mathbf{2} \leftarrow \mathcal{D}_{\mathbb{Z}^m,s_1,0}$.
4) Call $Sk_s \leftarrow \mathbf{Transform}(SK_{L_{ps}}, \mathbf{L_{ps}}, Att_s)$.
5) Sample $\sigma_\mathbf{1} \leftarrow \mathbf{SamplePre}(\mathbf{A_{Att_s}}, Plc_s, Sk_s, \upsilon - \mathbf{D}_\upsilon\sigma_\mathbf{2}, s_1)$, and compose $\sigma = (\sigma_\mathbf{1}, \sigma_\mathbf{2})$.
6) Compute $r_2 = H_1(\sigma)$.
7) Choose $\mathbf{t_1} \leftarrow D_{\mathbb{Z}^m,s,0}$ and $\mathbf{e_0} \leftarrow \mathcal{D}_{\mathbb{Z}^m,s_1,0}$.
8) Compute $\mathbf{t_g} = H_N(\mathbf{t_1}, r_2)$.
9) Choose a matrix $\mathbf{R} \xleftarrow{\$} \{-1,1\}^{m\times m}$ and a vector $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$.
10) Compute $\mathbf{c_1} = \mathbf{A_{Att_r}}^t\mathbf{s} + (\sigma, \mathbf{e_0})^t$, where $Att_r$ is a selected attribute set satisfying $Plc_r$;[1]
11) Compute $\mathbf{c_2} = (H(\mathbf{t_g})\mathbf{B} + \mathbf{C})^t\mathbf{s} + \mathbf{R}^t\mathbf{e_0}$.
12) Compute $\bar{c} = H_2(\sigma, \mathbf{s}, \mathbf{R}^t\mathbf{e_0}) \oplus (\mu, r_1, r_2)$, where the operator $\oplus$ denotes Exclusive-Or operation.[2]
13) Compute $\mathbf{c}' = H_3(\sigma, \mathbf{s}, \mathbf{R}^t\mathbf{e_0}) \oplus \mathbf{t_1}$.
14) Output $c = (\mathbf{t_g}, \mathbf{c_1}, \mathbf{c_2}, \mathbf{c}', \bar{c})$ as the ciphertext.

- UnSigncrypt$(c, Att_r, \mathbf{L_{pr}}, SK_{L_{pr}}, Att_s)$: On input a ciphertext $c$, the receiver's attribute set $Att_r$, the policy matrix $\mathbf{L_{pr}}$ satisfied by $Att_r$, the private key $SK_{L_{pr}}$ corresponding to $\mathbf{L_{pr}}$, and the sender's attribute $Att_s$, the receiver does:
  1) Call $\mathbf{Transform}(SK_{L_{pr}}, \mathbf{L_{pr}}, Att_r)$ to obtain private key $Sk_r$ for the attribute set $Att_r$, if the private key has not been computed previously and $Att_r$ satisfies $\mathbf{L_{pr}}$.
  2) Parse $c$ as $c = (\mathbf{t_g}, \mathbf{c_1}, \mathbf{c_2}, \mathbf{c}', \bar{c})$.
  3) Obtain $(\sigma, \mathbf{e_0})$ by computing $(\mathbf{T_r}^t)^{-1}(\mathbf{T_r}^t\mathbf{c_1} mod\ q)$.
  4) Solve equation $\mathbf{A_{Att_r}}^t\mathbf{s} = \mathbf{c_1} - (\sigma, \mathbf{e_0})^t$ to get $\mathbf{s}$.
  5) Compute $(\mu, r_1, r_2) = H_2(\sigma, \mathbf{s}, c_2 - (H(\mathbf{t_g})\mathbf{B} + \mathbf{C})^t\mathbf{s}) \oplus \bar{c}$.
  6) If $r_2 = H_1(\sigma)$, continue. Otherwise, output $\perp$ and abort.
  7) Compute $\mathbf{t_1} = H_3(\sigma, \mathbf{s}, c_2 - (H(\mathbf{t_g})\mathbf{B} + \mathbf{C})^t\mathbf{s}) \oplus \mathbf{c}'$.
  8) If $\mathbf{t_g} = H_N(\mathbf{t_1}, r_2)$ holds, continue. Otherwise output $\perp$ and abort.
  9) Compute $\upsilon = H_0(r_1, \mu, Att_s, \mathbf{L_{ps}}, Plc_r)$.
  10) Compute $\mathbf{D}_\upsilon = \sum_{i=0}^{\iota'}(-1)^{\upsilon[i]}\mathbf{D_i}$.
  11) If $\|\sigma\| \le s_1\sqrt{(\varrho+2)m}$ and $[\mathbf{A_{Att_s}}\|\mathbf{D}_\upsilon]\sigma = \upsilon$, output $\mu$. Otherwise, output $\perp$.

**Transform**$(\mathbf{K}, \mathbf{L}, As)$: Generate a private key $\mathbf{K^{(2)}}$ for the attribute set $As$.
Input: policy matrix $\mathbf{L}$, private key $\mathbf{K}$ for $\mathbf{L}$, attribute set $As$.
Output: private key $\mathbf{K^{(2)}}$ for attribute set $As$.

1) If $As$ dose not satisfy $\mathbf{L}$, return $\perp$.
2) Find an appropriate $\mathbf{g} \in \mathbb{Z}^k$ satisfying the following constraints.
   (1) If $i \notin As$ then $\mathbf{g}_i = 0$. Otherwise, $\mathbf{g}_i \neq 0$ and $\mathbf{g}_i$ is as small as possible.
   (2) Compute $(d\|0^\lambda) = \mathbf{g}^t\mathbf{L}$ to get some small integer $d$. Here $k$, $\lambda$ denote the number of all attributes and the columns of $\mathbf{L}$, respectively.
3) Extract a sub-matrix $\mathbf{K^{(1)}} \in \mathbb{Z}^{\rho m\times\rho m}$ from $\mathbf{K}$ by the following procedure.
   (1) If $i \notin As$, remove the corresponding rows of $\mathbf{K}$ for the attribute $i$. Then, obtain a matrix $\hat{\mathbf{K}} \in \mathbb{Z}^{(\rho+1)m\times(k+1)m}$.
   (2) Remove randomly $(k-\rho)m$ columns from $\hat{\mathbf{K}}$ to get $\mathbf{K^{(1)}}$.
4) Return $\mathbf{K^{(2)}} = (diag(\mathbf{g}_{i_1}, \mathbf{g}_{i_2}, \cdots, \mathbf{g}_{i_\rho}, d) \otimes \mathbf{I_m})\mathbf{K^{(1)}}$ as the secret key for attribute set $As$.

### C. CORRECTNESS AND PARAMETERS SETTING

First, we show that a valid private key of the attribute set is obtained in the algorithm **Transform** by the following two facts. Firstly,

$$\begin{aligned}\mathbf{P} &= \mathbf{A^{(1)}}\mathbf{K^{(2)}}\\ &= [\mathbf{A_{i_1}}\|\cdots\|\mathbf{A_{i_\rho}}\|\mathbf{A_0}]((diag(\mathbf{g}_{i_1}, \cdots, \mathbf{g}_{i_\rho}, d) \otimes \mathbf{I_m})\mathbf{K^{(1)}})\\ &= [\mathbf{g}_{i_1}\mathbf{A_{i_1}}\|\cdots\|\mathbf{g}_{i_\rho}\mathbf{A_{i_\rho}}\|d\mathbf{A_0}]\mathbf{K^{(1)}}\end{aligned}$$

On the one hand, $\mathbf{g}_i = 0$, when $i \notin As$. On the other hand, $\mathbf{K^{(1)}}$ is the sub-matrix of $\mathbf{K}$ corresponding to the columns of $As \cup \{k+1\}$. Therefore, $\mathbf{P}$ is a sub-matrix of $\mathbf{P^{(1)}}$, where $\mathbf{P^{(1)}} = [\mathbf{g}_1\mathbf{A_1}\|\mathbf{g}_2\mathbf{A_2}\|\cdots\|\mathbf{g}_k\mathbf{A_k}\|d\mathbf{A_0}]\mathbf{K}$. And, $\mathbf{P^{(1)}} = [\mathbf{g}_1\mathbf{A_1}\|\mathbf{g}_2\mathbf{A_2}\|\cdots\|\mathbf{g}_k\mathbf{A_k}\|d\mathbf{A_0}\|\mathbf{0}]\mathbf{K^{(3)}}$, where $\mathbf{0}$ is the zero matrix with dimension $n \times \lambda m$ and $\mathbf{K}$ is the uper sub-matrix of $\mathbf{K^{(3)}} \in \mathbb{Z}^{(k+\lambda+1)m\times(k+1)m}$, where $\mathbf{K^{(3)}}$ is the leftmost sub-matrix of $\mathbf{K}'$ and $\mathbf{K}'$ is the trapdoor of $\Lambda^\perp(\mathbf{M})$. As a result, $\mathbf{P^{(1)}}$ is the sub-matrix of $\mathbf{P^{(2)}}$.

$$\begin{aligned}\mathbf{P^{(2)}} &= [\mathbf{g}_1\mathbf{A_1}\|\mathbf{g}_2\mathbf{A_2}\|\cdots\|\mathbf{g}_k\mathbf{A_k}\|d\mathbf{A_0}\|\mathbf{0}]\mathbf{K}\\ &= ((\mathbf{g}_1\|\mathbf{g}_2\|\cdots\|\mathbf{g}_k) \otimes \mathbf{I_n})\mathbf{M}\mathbf{K}\\ &= \mathbf{0}\end{aligned}$$

As a result, $\mathbf{P} = \mathbf{0}$. Secondly, because

$$\begin{aligned}\|\mathbf{K^{(2)}}\| &= \|(diag(\mathbf{g}_{i_1}, \mathbf{g}_{i_2}, \cdots, \mathbf{g}_{i_\rho}, d) \otimes \mathbf{I_m})\mathbf{K^{(1)}}\|\\ &\le \max\{\mathbf{g}_{i_1}, \mathbf{g}_{i_1}, \cdots, \mathbf{g}_{i_\rho}, d\}\|\mathbf{K^{(1)}}\|.\end{aligned}$$

$\|\mathbf{K^{(2)}}\|$ is small. Therefore, $\mathbf{K^{(2)}}$ is the private key for $As$.

Second, the correct $(\sigma, \mathbf{e_0})^t$ is obtained in step 3 of unsigncryption.

$$\begin{aligned}&(\mathbf{T_r}^t)^{-1}(\mathbf{T_r}^t\mathbf{c_1} \mod q)\\ &\equiv (\mathbf{T_r}^t)^{-1}(\mathbf{T_r}^t\mathbf{A_{Att_r}}^t\mathbf{s} + \mathbf{T_r}^t(\sigma, \mathbf{e_0})^t \mod q)\\ &\equiv (\mathbf{T_r}^t)^{-1}(\mathbf{T_r}^t(\sigma, \mathbf{e_0})^t)\\ &= (\sigma, \mathbf{e_0})^t\end{aligned}$$

Then, the correct vector $\mathbf{s}$ can be obtained in step 4 of unsigncryption, similarly $(\mu, r_1, r_2)$ in step 5.

[1] For simplicity, we suppose that the number of the sender's attributes equals that of the receiver's. When the former is bigger, the redundant part can be hidden in the ciphertext $\mathbf{c}'$. On the contrary, the lacking error vectors can be chosen from the same Gaussian distribution.

[2] In fact, the length of the operation result is determined by the shorter one of the two operands, i.e. $(\mu, r_1, r_2)$.

Third, $\|\sigma\| \leq s_1\sqrt{(\varrho+2)m}$ holds due to ***Proposition*** 10 and ***Proposition*** 3, and $[\mathbf{A_{Att_s}}\|\mathbf{D}_\upsilon]\sigma = \nu$ naturally holds (see the analysis under ***Proposition*** 10). In summary, this scheme can unsigncrypt correctly. Certainly, the following requirements are needed.

- The SIS problem must be hard. According to ***Proposition*** 2, $q \geq \beta\omega(\sqrt{n\log n})$.
- The LWE problem must be hard. According to ***Proposition*** 1, $\alpha q > 2\sqrt{n}$.
- **TrapGen** algorithm should work well. According to ***Proposition*** 6, $m > 6n\log q$.
- **SampleLeft** and **SampleRight** algorithm should work well. It needs $s_1$ to be large enough, $s_1 = O(\sqrt{n\log q})\omega(\log^{1.5} m)(k+1)m$, according to ***Proposition*** 7 and ***Proposition*** 8.

According to the above constraints, the parameters should be set as follows, where $n^\delta > \lceil\log q\rceil$, and $Q$ is the number for signature queries.

$$m = 6n^{1+\delta}, \quad s_1 = O(\sqrt{n\log q})\omega(\log^{1.5} m)(k+1)m,$$
$$s = O(\sqrt{n\log q})\omega(\sqrt{\log m}),$$
$$s_2 = O(\sqrt{(n\log q)})\omega(\sqrt{\log km}),$$
$$\alpha = s_1/(\sqrt{2}q), \quad s_3 = O(\sqrt{n\log q})\omega(\log m)\sqrt{(k+1)m},$$
$$q = \max\{kO(n\log q)\omega(\log^{1.5} m)m^2\omega(\sqrt{n\log n}), 2Q\}.$$

*Remark 1:* Note that this scheme also supports threshold policy, when the encode for the threshold policy is a Vandermonde matrix. Certainly, the parameters should be adjusted accordingly. Please refer to [34] for more details.

### D. SECURITY

*Theorem 1 (Confidentiality): In the standard model, if there is an inner PPT adversary who can attack the proposed signcryption scheme in **Game IND-sAtt-CCA2** with non-negligible advantage, then there is an algorithm that can solve the decision-LWE$_{q,\alpha}$ problem for $\alpha = O((n\log q)^{0.5})\omega(\log^{1.5} m)(k+1)m/q$.*

*Proof:* Because $H_2$ and $H_3$ are universal hash functions, $\bar{c}$ (resp. $\mathbf{c}'$) is statistically indistinguishable with the uniform distribution over $\{0,1\}^{(|\mu|+|r_1|+|r_2|)}$ (resp. $\mathbb{Z}_q^m$). Therefore, the distinguishable ability of the adversary comes from $\mathbf{c_1}, \mathbf{c_2}$. To complete the proof, a sequence of games is defined as follows:

- $G_0$: This is the real game.
- $G_1$: Change the method to generate public keys. Without loss of generality, suppose the measure of the challenge attribute set is $\varrho$ and the challenge attribute set is $As_r^* = \{1,2,\cdots,\varrho\}$. $\mathcal{C}$ queries LWE oracle to obtain a group of LWE instance $(\mathbf{y_i}, x_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ for $i \in [(k+1)m]$. If attribute $i \notin As$, it keeps $(\mathbf{A_i}, \mathbf{T_i}) \leftarrow \mathbf{TrapGen}(1^\varpi)$, $\mathbf{Z_i} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$. Otherwise, $\mathbf{A_i} = [\mathbf{y_{(i-1)*m+1}}\|\mathbf{y_{(i-1)*m+2}}\|\cdots\|\mathbf{y_{i*m}}]$, $(\mathbf{Z_i}, \mathbf{S_i}) \leftarrow \mathbf{TrapGen}(1^\varpi)$. $\mathbf{A_0} = [\mathbf{y_{k*m+1}}\|\mathbf{y_{k*m+2}}\|\cdots\|\mathbf{y_{(k+1)*m}}]$.
- $G_2$: Change the method to generate $\mathbf{B}, \mathbf{C}$: $(\mathbf{B}, \mathbf{T_B}) \leftarrow \mathbf{TrapGen}(1^\varpi)$, $\mathbf{t_g^*} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{R}^* \xleftarrow{\$} \{1,-1\}^{m\times m}$, $\mathbf{C} = \mathbf{A_0}\mathbf{R}^* - H(\mathbf{t_g^*})\mathbf{B}$.
- $G_3$: The hash function $H_N$ is replaced with a chameleon hash function (see ***Proposition*** 11) $H_{N'}$ where $\mathbf{N}' \in \mathbb{Z}_q^{n\times(m+\iota)}$. In addition, in the procedure to generate the challenge ciphertext, $\mathbf{t_1} \leftarrow \mathcal{D}_{\Lambda_q^{\mathbf{h}'}(\mathbf{N}'_1)}$ where $\mathbf{h}' = \mathbf{t_g^*} - \mathbf{N}'_\mathbf{0}\mathbf{r_2}$. However, $\mathbf{t_1} \leftarrow D_{\mathbb{Z}^m,s,0}$ in Game $G_0 \sim G_2$.
- $G_4$: Continue changing the way to generate the challenge ciphertext. Specifically, $\mathbf{c_1^*} = (\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x}_\varrho, \mathbf{x_k})$, $\mathbf{c_2^*} = \mathbf{R}^*\mathbf{x_k}$, where $\mathbf{x_i} = (x_{(i-1)*m+1}, x_{(i-1)*m+2}, \cdots, x_{i*m})$.

The correctness of this theorem is implied in the facts that the successive games are indistinguishable and the adversary is just facing an LWE instance in the last scheme. □

*Lemma 1:* The games $G_0, G_1$ are statistically indistinguishable. And in $G_1$, the challenger $\mathcal{C}$ can reply the private key queries and decryption queries.

*Proof:* First, one difference between $G_0$ and $G_1$ is the generation method for some matrices. In $G_0$, $i \in As$, $\mathbf{Z_i} \leftarrow \mathbf{TrapGen}$; while $\mathbf{Z_i} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ in $G_1$. The lattice parity checking matrix generated by **TrapGen** algorithm is within a negligible statistical distance from $U(\mathbb{Z}_q^{n\times m})$, according to ***Proposition*** 6. Hence, the matrices $\mathbf{Z_i}$s in the two games are statistically indistinguishable. In $G_1$, $i \in As$ and $\mathbf{A_i} = [\mathbf{y_{i*m+1}}, \mathbf{y_{i*m+2}}, \cdots, \mathbf{y_{(i+1)*m}}]$; while $\mathbf{A_i} \leftarrow \mathbf{TrapGen}$ in $G_0$. If the LWE instance is from $O_s$, $\mathbf{A_i}$ in the two games has identical distribution. If the LWE instance is from $O_\$$, the statistical distance between the distribution of $\mathbf{A_i}$s in the two games is negligible according to the property of *TrapGen* (see ***Proposition*** 6). Hence, the distribution of public matrices is statistically indistinguishable.

Second, in $G_1$, $\mathcal{C}$ can reply the private key queries for policies as Boyen13 [22] does. For the convenience of depiction, suppose the attributes used for the private key query are $d_1, d_2, \cdots, d_{\varrho'}$.

1) If the attribute set $As^*$ satisfies the policy $\mathbf{L}$, then reply $\perp$.
2) Construct policy matrix $\mathbf{L}$ with LW algorithm.
3) For the convenience of depiction, let the symbol "$\Leftrightarrow$" denote "if and only if". The attribute set $As$ does not satisfy the policy $Plc$. $\Leftrightarrow$ The space extended by the rows $\{d_1, d_2, \cdots, d_{\varrho'}\}$ of $\mathbf{L}$ does not contain $(1,0,\cdots,0)$. $\Leftrightarrow$ The space extended by the rows $\{d_1, d_2, \cdots, d_{\varrho'}\}$ of $\mathbf{L}''$ does not contain $(0,\cdots,0)$, where $\mathbf{L}''$ is the matrix obtained by deleting the leftmost column of $\mathbf{L}$. $\Leftrightarrow$ The rows $\{d_1, d_2, \cdots, d_{\varrho'}\}$ of $\mathbf{L}''$ constitute a matrix with row full rank. $\Leftrightarrow$ The rows $\{d_1, d_2, \cdots, d_{\varrho'}\}$ of $\mathbf{L}''$ include at least a full rank square sub-matrix. W.l.o.g, suppose the full rank square sub-matrix corresponds to the columns $\{c_1, c_2, \cdots, c_{\varrho'}\}$.
4) If $i \in \{c_1, c_2, \cdots, c_{\varrho'}\}$, $(\mathbf{Z_i}, \mathbf{S_i}) \leftarrow \mathbf{TrapGen}(1^\varpi)$, for $i \in [\lambda]$. Otherwise, $\mathbf{Z_i} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$.
5) Let $\mathbf{L}'_{(\mathbf{i})} = (\mathbf{L}_{1,i}, \mathbf{L}_{2,i}, \cdots, \mathbf{L}_{\varrho',i})$ for $i = 1,2,\cdots,\lambda$, where $\mathbf{L}'_{(\mathbf{i})}$ denotes the $i$-th column of matrix $\mathbf{L}'$. Then, $\mathbf{L_{zl}} = [\mathbf{L}'_{(c_1)} \otimes \mathbf{Z_{c_1}}\|\mathbf{L}'_{(c_2)} \otimes \mathbf{Z_{c_2}}\|\cdots\|\mathbf{L}'_{(c_{\varrho'})} \otimes \mathbf{Z_{c_{\varrho'}}}]$ is a sub-matrix of $\mathbf{M}$ (see step 3 of algorithm Extract).
6) It is easy to check that $diag(\mathbf{S_{c_1}}, \mathbf{S_{c_2}}, \cdots, \mathbf{S_{c_{\varrho'}}})$ is a trapdoor of $\Lambda^\perp(\mathbf{L_{zl}})$.

7) Exchange the columns in $\mathbf{M}$ corresponding to $diag(\mathbf{A_{d_1}}, \mathbf{A_{d_2}}, \cdots, \mathbf{A_{d_{\varrho'}}})$ and that corresponding to $\mathbf{L_{zl}}$ to get a matrix $\mathbf{P}$, where $\mathbf{M} = [diag(\mathbf{A_1}, \mathbf{A_2}, \cdots, \mathbf{A_k}) \| \mathbf{L}_{(0)} \otimes \mathbf{A_0} \| \mathbf{L}_{(1)} \otimes \mathbf{Z_1} \| \cdots \| \mathbf{L}_{(\lambda)} \otimes \mathbf{Z}_\lambda]$. Let $\mathbf{P_l}$ denote the diagonal sub-matrix constituted by the leftmost $km$ columns of $\mathbf{P}$, and $\mathbf{P_r}$ the sub-matrix constituted by the remainder columns.
8) Construct a block diagonal matrix $\mathbf{S_T} \in \mathbb{Z}^{km \times km}$. If $i \notin \{d_1, d_2, \cdots, d_{\rho'}\}$, set the $i$-th block to be $\mathbf{T_i}$; otherwise, set it to be $\mathbf{S_{c_k}}$. Here, $\mathbf{S_{c_k}}$ is the first key matrix not be appointed. Obviously, $\mathbf{S_T}$ is a trapdoor for $\Lambda^{\perp}(\mathbf{P_l})$.
9) Extend the basis

$$\mathbf{S'_{Tr}} \leftarrow \mathbf{SampleBaisLeft}(\mathbf{P_l}, \mathbf{P_r}, \mathbf{S_T}, s_2).$$

10) Exchange the $1 \sim \varrho' m$ columns with the $(k + c_1)m + 1 \sim (k + c_1 + 1)m$, $(k + c_2)m + 1 \sim (k + c_2 + 1)m$, $\cdots$, $(k + c_{\varrho'})m + 1 \sim (k + c_{\varrho'} + 1)m$ columns in $\mathbf{S'_{Tr}}$ to get $\mathbf{S_{Tr}}$. Give $(\mathbf{S_{Tr}}, \mathbf{L})$ as the secret key for policy $Plc$.

In fact, the bases in $G_0$ and $G_1$ are both obtained by calling algorithm **SampleBaisLeft**. And the Gaussian parameters used are identical. According to ***Proposition*** 9, the two bases are statistically indistinguishable. Given a policy, $\mathcal{C}$ can get a corresponding private key with the above method. It is natural that $\mathcal{C}$ can reply the decryption query with this private key. □

*Lemma 2:* The games $G_1, G_2$ are statistically indistinguishable. In $G_2$, $\mathcal{C}$ can reply the unsigncryption queries.

*Proof:* According to the uniform distribution property of the parity checking matrix generated by **TrapGen**, the $\mathbf{B}$ in $G_1$ is statistically indistinguishable from the $\mathbf{B}$ in $G_2$. According to Lemma 13 of [26], the $\mathbf{C}$ is also statistically indistinguishable from that in $G_2$.

If the attribute set $As$ does not satisfy the policy $\mathbf{L}$, $\mathcal{C}$ replies with $\perp$. Otherwise, $\mathcal{C}$ replies the decryption queries as follows. Without loss of generality, let $Att = \{i_1, i_2, \cdots, i_\rho\}$.

1) If $\mathbf{t_g} = \mathbf{t_g^*}$, $\mathcal{C}$ replies $\perp$. Note that the public key corresponding to the ciphertext is $\mathbf{P}_{pub} = [\mathbf{A_{i_1}} \| \mathbf{A_{i_2}} \| \cdots \| \mathbf{A_{i_\varrho}} \| \mathbf{A_0} \| H(\mathbf{t_g^*})\mathbf{B} + \mathbf{C}] = [\mathbf{A_{i_1}} \| \mathbf{A_{i_2}} \| \cdots \| \mathbf{A_{i_\varrho}} \| \mathbf{A_0} \| \mathbf{A_0}\mathbf{R}^* + H(\mathbf{t_g} - \mathbf{t_g^*})\mathbf{B}]$.
2) $\mathbf{T}_{A0B} \leftarrow \mathbf{SampleBasisRight}(\mathbf{A_0}, \mathbf{B}, H(\mathbf{t_g} - \mathbf{t_g^*}), \mathbf{R}^*, \mathbf{T_B}, s_4)$, $s_4 = O(\sqrt{n \log q})\omega(\sqrt{\log m})\sqrt{m}$. That is, $\mathbf{T}_{A0B}$ is a trapdoor for $\Lambda^{\perp}(\mathbf{A'})$, where $\mathbf{A'} = [\mathbf{A_0} \| \mathbf{A_0}\mathbf{R}^* + H(\mathbf{t_g} - \mathbf{t_g^*})\mathbf{B}]$.
3) Parse $\mathbf{c_1} = (\mathbf{c'_1}, \mathbf{c''_1}) \in \mathbb{Z}_q^{\varrho m} \times \mathbb{Z}_q^m$. Obviously, $\mathbf{c'_1} = \mathbf{A_{Att}}^t \mathbf{s} + \sigma$, $\mathbf{c''_1} = \mathbf{A_0}^t \mathbf{s} + \mathbf{e_0}$, for some $\mathbf{s}, \sigma, \mathbf{e_0}$.
4) Compose a new ciphertext $(\mathbf{c''_1}, \mathbf{c_2})$. Obviously, $(\mathbf{c''_1}, \mathbf{c_2}) = (\mathbf{A_0}^t \mathbf{s} + \mathbf{e_0}, (\mathbf{A_0}\mathbf{R}^* + H(\mathbf{t_g} - \mathbf{t_g^*})\mathbf{B})^t \mathbf{s} + \mathbf{R}^* \mathbf{e_0}) = [\mathbf{A_0} \| \mathbf{A_0}\mathbf{R}^* + H(\mathbf{t_g} - \mathbf{t_g^*})\mathbf{B}]^t \mathbf{s} + (\mathbf{e_0}, \mathbf{R}^{*t} \mathbf{e_0})$.
5) Decrypt the ciphertext $(\mathbf{c''_1}, \mathbf{c_2})$ with the trapdoor $\mathbf{T}_{A0B}$ to get $\mathbf{s}, \mathbf{e_0}, \mathbf{R}^{*t} \mathbf{e_0}$.
6) Compute $\sigma = \mathbf{c'_1} - \mathbf{A_{Att}}^t \mathbf{s}$.
7) Because $\mathcal{C}$ has $\mathbf{s}, \sigma, \mathbf{e_0}$, it can normally execute the subsequent decryption procedure, namely steps 5. $\sim$ 10. in **Unsigncrypt**.

The only difference in replying unsigncryption queries in $G_2$ and $G_1$ is that $\mathcal{C}$ cannot unsigncrypt when $\mathbf{t_g} = \mathbf{t_g^*}$ in $G_2$. Before the challenge ciphertext is generated, $\mathbf{t_g^*} \in \mathbb{Z}_q^n$ is hidden from the adversary $\mathcal{A}$. Even not considering the collision of the hash function $H_N$, the probability that $\mathbf{t_g}$ generated normally satisfies $\mathbf{t_g} = \mathbf{t_g}^*$ is $q^{-n}$, which is negligible. After the challenge ciphertext is published, $\mathcal{A}$ knows $\mathbf{t_g^*}$. If $\mathcal{A}$ can normally generate $\mathbf{t_g}$ to satisfy $\mathbf{t_g} = \mathbf{t_g^*}$, then $\mathcal{A}$ can find a collision for the hash function $H_N$ or $H_1$. Hence, this probability is negligible. Therefore, $G_1$ and $G_2$ are statistically indistinguishable. □

*Lemma 3:* The games $G_2, G_3$ are statistically indistinguishable.

*Proof:* In $G_2$, $\mathbf{N} = [\mathbf{N_l} \| \mathbf{N_r}] \in \mathbb{Z}_q^{n \times (m+\iota)}$ where $\mathbf{N_l} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, $\mathbf{N_r} \xleftarrow{\$} \mathbb{Z}_q^{n \times \iota}$. In $G_3$, $\mathbf{N} = [\mathbf{N_l} \| \mathbf{N_r}] \in \mathbb{Z}_q^{n \times (m+\iota)}$ where $\mathbf{N_l} \leftarrow \mathbf{TrapGen}$, $\mathbf{N_r} \xleftarrow{\$} \mathbb{Z}_q^{n \times \iota}$. According to the property of **TrapGen** (see ***Proposition*** 6), the two hash functions are statistically indistinguishable. On the one hand, in $G_3$, the random vector $\mathbf{t_1}$ chosen from $\mathcal{D}_{\Lambda_q^{\mathbf{h}'}(\mathbf{N}'_1)}$ is used only once in the challenge ciphertext. On the other hand, this $\mathbf{t_1}$ also satisfies $\|\mathbf{t_1}\| \leq s\sqrt{m}$ with overwhelming probability, according to item 2 of ***Proposition*** 3. Hence, this $\mathbf{t_1}$ cannot be distinguished from the one normally generated. □

*Lemma 4:* The games $G_3, G_4$ are computationally indistinguishable. In $G_4$, the challenge ciphertext is exactly an LWE instance.

*Proof:* If the LWE instance given by LWE oracle comes from $O_\$$, the games $G_3, G_4$ are computationally indistinguishable due to LWE hardness (see ***Proposition*** 1). If the LWE instance is from $O_s$, the game $G_4$ is a case of $G_3$ due to the following fact. In other words, $G_4$ and $G_3$ have identical distribution.

$$\begin{aligned}
\mathbf{c_1^*} &= [\mathbf{A_1} \| \mathbf{A_2} \| \cdots \| \mathbf{A_\varrho} \| \mathbf{A_0}]^t \mathbf{s} + (\mathbf{e_1}, \mathbf{e_2}, \cdots, \mathbf{e_\varrho}, \mathbf{e_0}) \\
&= \mathbf{A_1}^t \mathbf{s} + \mathbf{e_1} \| \mathbf{A_2}^t \mathbf{s} + \mathbf{e_2} \| \cdots \| \mathbf{A_\varrho}^t \mathbf{s} + \mathbf{e_\varrho} \| \mathbf{A_0}^t \mathbf{s} + \mathbf{e_0} \\
&= (\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_\varrho}, \mathbf{x_{k+1}}) \\
\mathbf{c_2^*} &= (H(\mathbf{t_g^*})\mathbf{B} + \mathbf{C})^t \mathbf{s} + \mathbf{R}^* \mathbf{e_0} \\
&= (H(\mathbf{t_g^*})\mathbf{B} + \mathbf{A_0}\mathbf{R}^* - H(\mathbf{t_g^*})\mathbf{B})^t \mathbf{s} + \mathbf{R}^* \mathbf{e_0} \\
&= \mathbf{R}^{*t}(\mathbf{A_0}^t \mathbf{s} + \mathbf{e_0}) \\
&= \mathbf{R}^{*t} \mathbf{x_{k+1}}
\end{aligned}$$

□

*Theorem 2 (EUF-sAtt-CMA): In the standard model, if there is a PPT inner adversary who can forge a signature in* ***Game EUF-sAtt-CMA*** *with non-negligible probability, then there is an efficient algorithm to solve $SIS_{q,\beta}$ for $\beta = C\iota' m s_1$.*

*Proof:*

- Initial: $\mathcal{F}$ submits an attribute set $As_s^*$ that it wants to attack. $\mathcal{C}$ builds matrices as follows. 1) Generate trapdoor $(\mathbf{F}, \mathbf{T_F}) \leftarrow \mathbf{TrapGen}(1^\varpi)$. 2) Compute $\mathbf{D_i} = \mathbf{A_0}\mathbf{R_i} + h_i\mathbf{F}$ for $0 \leq i \leq \iota'$, where $\mathbf{A_0} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, $\mathbf{R_i} \leftarrow D_{\mathbb{Z},s}^{m \times m}$, $h_i \xleftarrow{\$} \mathbb{Z}_q$ but $h_0 = 1$. According to ***Proposition*** 5, the $\mathbf{D_i}$ generated by this method and $\mathbf{D_i} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$ are statistically indistinguishable. 3) Choose $\mathbf{x_1} \leftarrow D_{\mathbb{Z}^{\varrho m}, s_1}$, $\mathbf{x_2} \leftarrow D_{\mathbb{Z}^m, s_1}$. Compute $\mathbf{y} = [\mathbf{A_1} \| \mathbf{A_2} \| \cdots, \mathbf{A_\varrho} \| \mathbf{A_0}](\mathbf{x_1}, \mathbf{x_2})$. 4) The generation

methods for the other matrices (e.g. $\mathbf{B}$, $\mathbf{C}$, $\mathbf{A_i}$, $\mathbf{Z_j}$ for $0 \leq i \leq k, j \in [k]$) are the same as that in $G_1$ of ***Theorem*** 1, respectively.

- Queries:
  - private key queries: $\mathcal{C}$ deals with the private key queries as in $G_1$ of ***Theorem*** 1.
  - signcrypt queries: There are two cases:
    (1) The sender's attribute set does not satisfy the policy $\mathbf{L_{ps}}$. $\mathcal{C}$ replies $\perp$. (2) The sender's attribute set satisfies the policy $\mathbf{L_{ps}}$. $\mathcal{F}$ does as follows. 1) Choose $r_1 \xleftarrow{\$} \{0,1\}^{\iota}$ and compute $\upsilon = H_0(r_1, \mu, Att_s, \mathbf{L_{ps}}, Plc_r)$. 2) Compute $\mathbf{D}_\upsilon = \sum_{i=1}^{\iota'}(-1)^{\upsilon[i]}\mathbf{D_i} = \mathbf{A_0R} + h\mathbf{F}$, where $\mathbf{R} = \sum_{i=1}^{\iota'}(-1)^{\upsilon[i]}\mathbf{R_i}$, $h = \sum_{i=1}^{\iota'}(-1)^{\upsilon[i]}h_i$. If $h = 0$ goto step 1). 3) Choose $\sigma_1 \leftarrow \mathcal{D}_{\mathbb{Z}^{\varrho m}, s_1}$ and compute $\nu' = \nu - [\mathbf{A_{i_1}}\|\mathbf{A_{i_2}}\|\cdots,\mathbf{A_{i_{\varrho'}}}]\sigma'$. 4) Get Basis $\mathbf{Te} \leftarrow \mathbf{SampleBasisRight}(\mathbf{A_0}, \mathbf{F}, h, \mathbf{R}, \mathbf{T_B}, s_5)$, where $s_5 = O(\sqrt{(n\log q)})\ \omega(\sqrt{\log m})\sqrt{2m}$; 5) $\sigma_2 \leftarrow \mathbf{SamplePre}([\mathbf{A_0}\|\mathbf{A_0R} + h\mathbf{F}], \mathbf{Te}, \nu', s_1)$. 6) According to [35, Th. 3.4], the sample $(\sigma_1, \sigma_2)$ obeys the distribution $\mathcal{D}_{\Lambda_q^{\upsilon}(\mathbf{A}'), s_1}$. Output $(\sigma_1, \sigma_2)$ as the signature.
  - unsigncrypt queries: There are two cases:
    (1) If the receiver's attribute set does not satisfy the policy $\mathbf{L_{pr}}$, $\mathcal{C}$ replies $\perp$.
    (2) The receiver's attribute set $As_r \neq As_s^*$ satisfies the policy $\mathbf{L_{pr}}$, then $\mathcal{F}$ can unsigncrypt with the method in $G_2$ of ***Theorem*** 1.
- Forge: Finally, $\mathcal{F}$ outputs a valid forgery signature $(r_1, \sigma)$ for some message $\mu$ under the attribute set $As^*$, where $\sigma \in \mathbb{Z}^{(\varrho+2)m}$. For simplicity, express $\sigma$ as $(\sigma_1, \sigma_2, \sigma_3) \in \mathbb{Z}^{\varrho m} \times \mathbb{Z}^m \times \mathbb{Z}^m$.
  1) Compute $\upsilon = H_0(r_1, \mu, As_s^*, \mathbf{L_{ps}}, Plc_r)$.
  2) Compute $h = \sum_{i=1}^{\iota'}(-1)^{\upsilon[i]}h_i$. If $h \neq 0$, abort.
  3) Compute $\mathbf{R}' = \sum_{i=1}^{\iota'}(-1)^{\upsilon[i]}\mathbf{R_i}$. It is easy to check that $\mathbf{D}_\upsilon = \mathbf{A_0R}'$.
  4) Compute $\sigma' = (\mathbf{x_1}, \mathbf{x_2}) - (\sigma_1, \sigma_2 + \mathbf{R}'\sigma_3)$. Return $\sigma'$ as a solution for the SIS problem. The reason is shown in the next lemma.

□

*Lemma 5:* The vector $\sigma'$ obtained above is a valid solution for $\mathrm{SIS}_{q,\beta}$ problem with overwhelming probability, for $\beta = kO(\sqrt{n\log q})\omega(\log^{1.5} m)m^2$. And the probability that $\mathcal{C}$ gets the solution is non-negligible.

*Proof:* At first, due to

$$\begin{cases} \mathbf{y} = [\mathbf{A_1}\|\mathbf{A_2}\|\cdots\|\mathbf{A}_\varrho\|\mathbf{A_0}](\mathbf{x_1}, \mathbf{x_2}) \\ \mathbf{y} = [\mathbf{A_1}\|\mathbf{A_2}\|\cdots\|\mathbf{A}_\varrho\|\mathbf{A_0}\|\mathbf{D}_\upsilon](\sigma_1, \sigma_2, \sigma_3), \end{cases}$$

$$[\mathbf{A_1}\|\mathbf{A_2}\|\cdots\|\mathbf{A}_\varrho\|\mathbf{A_0}](\mathbf{x_1}, \mathbf{x_2}) = [\mathbf{A_1}\|\mathbf{A_2}\|\cdots\|\mathbf{A}_\varrho\|\mathbf{A_0}\|\mathbf{A_0R}](\sigma_1, \sigma_2, \sigma_3)$$

$$[\mathbf{A_1}\|\mathbf{A_2}\|\cdots\|\mathbf{A}_\varrho\|\mathbf{A_0}]((\mathbf{x_1}, \mathbf{x_2}) - (\sigma_1, \sigma_2 + \mathbf{R}\sigma_3)) = \mathbf{0}.$$

$$[\mathbf{A_1}\|\mathbf{A_2}\|\cdots\|\mathbf{A}_\varrho\|\mathbf{A_0}]\sigma' = \mathbf{0},$$

for $\sigma' = (\mathbf{x_1}, \mathbf{x_2}) - (\sigma_1, \sigma_2 + \mathbf{R}\sigma_3)$.

Second,

$$\begin{aligned} \beta &= \|\sigma'\| = \|(\mathbf{x_1}, \mathbf{x_2}) - (\sigma_1, \sigma_2 + \mathbf{R}\sigma_3)\| \leq \|(\mathbf{x_1}, \mathbf{x_2})\| \\ &\quad + \|(\sigma_1, \sigma_2 + \mathbf{R}\sigma_3)\| \\ &\leq \|(\mathbf{x_1}, \mathbf{x_2})\| + \|\sigma_1\| + \|\sigma_2\| + \|\mathbf{R}\|\|\sigma_3\| \\ &\leq s_1\sqrt{(\varrho+1)m} + s_1\sqrt{\varrho m} + C\iota'\sqrt{m+m}s_1\sqrt{m} \\ &\leq C\iota' \cdot s_1 \cdot m \leq C\iota' \cdot O(\sqrt{n\log q})\omega(\log^{1.5} m)(k+1)m^2 \\ &= kO(\sqrt{n\log q})\omega(\log^{1.5} m)m^2 \end{aligned}$$

Here, the third inequality holds due to ***Proposition*** 4,3 and 10.

Third, when $q$ is bigger than the double of the number of queries $Q$, $\mathcal{C}$ will get this solution with probability more than $2/3$, according to Lemma 26 and 27 of [36]. □

## V. PERFORMANCE AND VARIANT

### A. PERFORMANCE ANALYSIS

In this section, let us compare the performance of the proposed scheme with the mechanism of signature and then encryption. For clarity, we give the universal conclusion for the computational overhand for pre-image sampling algorithm, firstly.

*Lemma 6:* When the parity check matrix belongs to $\mathbb{Z}_q^{n\times m}$, the cost of the pre-image sampling algorithm is roughly $m^2 + 2mn$ $\mathbb{Z}_q^{\times}$ with some optimization.

*Proof:* The pre-image algorithm involves three steps, namely solving equation, SampleD and vector addition. The cost of vector addition is much less than the other two steps, and it is ignored.

The solving equation group $\mathbf{Ax} = \mathbf{y}$ involves Gaussian elimination and back substitution operations. In the Gaussian elimination, a triangle matrix $\mathbf{B}$ multiplies the coefficient matrix $\mathbf{A}$ to obtain trapezoid result matrix $\mathbf{D}$, namely, $\mathbf{BA} = \mathbf{D}$. The repeating elimination operation can be avoided by storing $\mathbf{B}$ and $\mathbf{D}$. Given a new syndrome $\mathbf{y}$, it only needs to compute $\mathbf{y_1} = \mathbf{By}$ and execute back substitution for $\mathbf{Dx} = \mathbf{y_1}$, due to $\mathbf{By} = \mathbf{y_1} = \mathbf{Dx} = \mathbf{BAx}$ and $\mathbf{y} = \mathbf{B}^{-1}\mathbf{By} = \mathbf{B}^{-1}\mathbf{BAx} = \mathbf{Ax}$. The cost of the multiplication $\mathbf{By}$ is about $n(n+1)/2$ $\mathbb{Z}_q^{\times}$. The back substitution is roughly $mn - (n-1)n/2$ $\mathbb{Z}_q^{\times}$. As a result, the cost to solve equation group is about $mn$ $\mathbb{Z}_q^{\times}$.

SampleD can be divided into the multiplications between the Gram-Schmidt vector of the basis and the evolutive vector of a solution, the inner products of Gram-Schmidt vector of the basis, the scalar multiplications for a small integer and a vector with small elements, and $m$ discrete Gaussian sampling (DGS). The magnitude of DGS is much less than the elementary operations in the other steps, so its cost is neglected. The overheads of the first three kinds of operations are $m^2$ $\mathbb{Z}_q^{\times}$, $m^2$ $\mathbb{Z}^{\times}$ and $m^2$ $\mathbb{Z}^{\times}$, respectively. The cost of the inner product operation is ignored by storing the value of the inner product. According to experiments, the cost of $\mathbb{Z}_q^{\times}$ is roughly 341.7 times of that of $\mathbb{Z}^{\times}$. Therefore, the cost of SampleD is about $m^2$ $\mathbb{Z}_q^{\times}$.

The above analysis can well support this lemma. □

*Remark 2:* In the StE approach, to guarantee the non-malleability of ciphertext, the ciphertext should be signed

with an unforgeable signature algorithm. If the public and private keys of the signature are published with PKI or identity mechanism, not only is the key size in the system too tremendous, but this will also leak the individual information of the signer. Hence, it is a better way to sign with the key of attributes. Based on this, the performance of the proposed signcryption scheme and that of StE scheme are compared as follows.

First, the key sizes are compared. In StE, no extra public keys and private keys are needed for the signature for ciphertext, according to Remark 2. Therefore, the public key size and private key size are identical in the two mechanisms, respectively.

Second, the computational costs are compared. In the signcryption, the computational cost mainly lies in steps 4, 5, 10 and 11. The step 4 extracts the private key corresponding to the attribute set. Not only can it be executed in advance, but also it needs to be run only once for the same attribute set under a policy. Hence, its computational cost in every signcryption is ignored. The cost of step 5 is $\rho^2m^2 + (\rho + 1)nm$ $\mathbb{Z}_q^\times$, according to Lemma 6. The costs of step 10 and 11 are $\rho$ $nm$ $\mathbb{Z}_q^\times$ and $nm$ $\mathbb{Z}_q^\times$, respectively. Therefore, the total cost of **Signcrypt** is $\rho^2m^2+2(\rho+1)nm$ $\mathbb{Z}_q^\times$. In StE, an extra signature is used to guarantee the non-malleability of ciphertext, and its overhead is $\rho^2m^2 + (\rho + 1)nm$ $\mathbb{Z}_q^\times$. Therefore, the total cost of the signature and encryption in StE is $2\rho^2m^2 + 3(\rho + 1)nm$ $\mathbb{Z}_q^\times$. That means the **Signcrypt** cost is less than 50% of that of StE. In the signcryption scheme, the computational cost of unsigncryption mainly lies in steps 3, 4, 5 and 11. Their costs are $(\varrho+1)^2m^2$ $\mathbb{Z}_q^\times+(\varrho+1)^2m^2$ $\mathbb{R}^\times$, $(\varrho + 1)$ $nm$ $\mathbb{Z}_q^\times$, $nm$ $\mathbb{Z}_q^\times$ and $(\varrho + 2)m$ $\mathbb{Z}^\times + (\varrho + 2)nm$ $\mathbb{Z}_q^\times$, respectively. The total cost of unsigncryption is $[(\varrho+1)^2m^2+(2\varrho + 4)nm]$ $\mathbb{Z}_q^\times + (\varrho + 1)^2m^2$ $\mathbb{R}^\times$. In StE, the verification cost of the extra signature is $(\varrho + 2)nm]$ $\mathbb{Z}_q^\times$. Therefore, the computational cost of decryption and verification for StE is $[(\varrho + 1)^2m^2 + (3\varrho + 6)nm]$ $\mathbb{Z}_q^\times + (\varrho + 1)^2m^2$ $\mathbb{R}^\times$. Our advantage is not obvious.

Third, the ciphertext sizes are compared. The ciphertext extension is defined as $\Im = |c| - |u|$, where $|c|$ and $|u|$ denote the length of ciphertext and that of plain text, respectively. The proportion of $\Im$ of the proposed scheme and StE is as follows. Especially, when $\varrho = 1, 2$, $\Re = \frac{2}{3}, \frac{5}{7}$, respectively.

$$\begin{aligned}\Re &= \frac{|\mathbf{c_1}| + |\mathbf{c_2}| + |\mathbf{c'}| + |c| - |u|}{|\mathbf{c_1}| + |\mathbf{c_2}| + |\mathbf{c'}| + |c| + |\sigma'| - |u|} \\ &= \frac{((\varrho + 1)m + m + m)\log q + \iota + \varsigma}{((\varrho + 1)m + m + m + 2m)\log q + \iota + \varsigma} \\ &\approx \frac{\varrho + 3}{\varrho + 5}\end{aligned}$$

### B. VARIANT

In fact, the computational cost of unsigncryption operation can be reduced greatly by replacing the public encryption section in the above scheme. In the variant, the steps of 11 and 12 are replaced with the following steps.

11(a). Compute $\mathbf{c_2} = (H(\mathbf{t_g})\mathbf{B} + \mathbf{C})^t\mathbf{s} + \mathbf{R}^t\mathbf{e_0}$.

11(b). Compute $\mathbf{c_3} = \mathbf{U}^t\mathbf{s} + H_2(\sigma)\lfloor q/2 \rfloor$.

12. Compute $c = G(H_2(\sigma)) \oplus (\sigma, \mu, r_1, r_2)$.

Here, $\mathbf{U} \in \mathbb{Z}_q^{n\times\iota}$ is a matrix from the public key, and $G$ is a random number generator. Then, the steps 3, 4 and 5 of unsigncryption algorithm in the above scheme are changed as follows:

3. Repeat $\mathbf{X}_i \leftarrow \mathbf{SamplePre}(\mathbf{A_{Att}}, sk_{Att}, \mathbf{U}_i, s_1)$ for $i \in [\iota]$;

4. Compute $\tilde{\sigma} = \mathbf{c_3} - \mathbf{X}^t\mathbf{c_1}$. If $\|\tilde{\sigma}_i\| \le \lfloor q/4 \rfloor$, set $\hbar_i = 0$. Otherwise, set $\hbar_i = 1$, for $i \in [\iota]$;

5. Compte $(\sigma, \mu, r_1, r_2) = G(\hbar) \oplus c$;

In the security reduction, the method to reply decryption queries is as follows.

1) Parse $\mathbf{c_1} = (\mathbf{c_1'}, \mathbf{c_1''}) \in \mathbb{Z}_q^{\varrho m} \times \mathbb{Z}_q^m$. Note that $\mathbf{c_1''} = \mathbf{A_0}^t\mathbf{s} + \mathbf{e_1}^t$.
2) Compose $(\mathbf{c_1''}, \mathbf{c_2})$. Note that $(\mathbf{c_1''}, \mathbf{c_2}) = [\mathbf{A_0}\|H(\mathbf{t_g})\mathbf{B} + \mathbf{C}]^t\mathbf{s} + (\mathbf{e_1}, \mathbf{Re_1})^t$
$= [\mathbf{A_0}\|\mathbf{A_0R} + (H(\mathbf{t_g}) - H(\mathbf{t_g^*}))\mathbf{B}]^t\mathbf{s} + (\mathbf{e_0}, \mathbf{R}^t\mathbf{e_1})$.
3) Run $\mathbf{T} \leftarrow \mathbf{SampleBasisRight}(\mathbf{A_0}, \mathbf{B}, (H(\mathbf{t_g}) - H(\mathbf{t_g^*})), \mathbf{R}, \mathbf{T_B}, s_5)$.
4) Repeat $\mathbf{X}_i \leftarrow \mathbf{SamplePre}([\mathbf{A_0}\|\mathbf{A_0R} + (H(\mathbf{t_g}) - H(\mathbf{t_g^*}))\mathbf{B}], \mathbf{T}, \mathbf{U}_i, s_1)$ for $i \in [\iota]$.
5) Compute $\tilde{\sigma} = \mathbf{c_3} - \mathbf{X}^t\mathbf{c_1}$. If $\|\tilde{\sigma}_i\| \le \lfloor q/4 \rfloor$ set $\hbar_i = 0$; otherwise set $\hbar_i = 1$ for $i \in [\iota]$.
6) Compte $(\sigma, \mu, r_1, r_2) = G(\hbar) \oplus c$.

Then, the other steps remain unchanged. Finally, $\mathcal{C}$ gives $\mu$ or $\perp$ as the reply.

In the **SignCrypt**, only the cost of 11.(b) is slightly big, namely $n\iota$ $\mathbb{Z}_q^\times$ for $\iota = 80$. Hence, the cost of **SignCrypt** is $\rho^2m^2 + 2(\rho + 1)nm + n\iota$ $\mathbb{Z}_q^\times \approx \rho^2m^2 + 2(\rho + 1)nm$ $\mathbb{Z}_q^\times$. In the **UnSigncrypt** algorithm, the computational cost mainly focuses on steps 3, 4 and 11. Their costs are $\iota[(\varrho + 1)^2m^2 + 2(\varrho + 1)nm]$ $\mathbb{Z}_q^\times$, $\iota(\varrho + 1)m$ $\mathbb{Z}_q^\times$, and $(\varrho + 2)nm$ $\mathbb{Z}_q^\times + (\varrho + 2)m$ $\mathbb{Z}^\times$, respectively. Because the operations in the step 3 of the unsigncryption have nothing to do with the ciphertext, this step can be pre-computed. Therefore, the cost of unsigncryption is $\iota(\varrho+1)m+(\varrho+2)nm$ $\mathbb{Z}_q^\times$. The total computational cost of verification, decryption and verification of StE is $\iota(\varrho + 1)m + 2(\varrho + 2)nm$ $\mathbb{Z}_q^\times$. That is, the computational overhead saves roughly 50% compared with the StE mechanisam.

**TABLE 1. Comparison between YW-ABSC and StE**

| Scheme | SignCrypt Sign & Enc | unSignCrypt Dec & Vfy | ciphertext size |
|---|---|---|---|
| YW-ABSC. | $\rho^2m^2 + (\rho + 1)nm$ | $(\varrho + 1)^2m^2$ | $(\varrho + 3)m \log q$ |
| StE. | $2\rho^2m^2 + 3(\rho + 1)nm$ | $(\varrho + 1)^2m^2$ | $(\varrho + 5)m \log q$ |

## VI. CONCLUSIONS

In this paper, an key policy ABSC scheme is put forward based on LWE and SIS hardness assumptions. For reducing the ciphertext size, Regev's encryption variant is used to directly hide the signature with big size. In the construction, a method is found to improve a non-semantic

secure encryption scheme to IND-CCA1 security by introducing an extra ciphertext with small size. Furthermore, the unforgeability of the signature for messages is reused so that the proposed ABSC is proved IND-CCA2 secure against inner adversary in the standard model. In the ABSC scheme, an ABS scheme based on lattice is constructed, which is proved EUF-CMA against inner adversary in the standard model. The theoretical analysis shows that the computational cost is reduced obviously, especially when the maximum measure of the minimum attribute set is not big. In addition, it is interesting to design an efficient ciphertext policy ABSC scheme from lattices. We defer it to the future work.

## REFERENCES

[1] K. Emura, A. Miyaji, and M. S. Rahman, "Dynamic attribute-based signcryption without random oracles," *Int. J. Adv. Comput. Technol.*, vol. 2, no. 3, pp. 199–211, 2012. doi: 10.1504/IJACT.2012.045589.

[2] J. Wei, X. Hu, and W. Liu, "Traceable attribute-based signcryption," *Secur. Commun. Netw.*, vol. 7, no. 12, pp. 2302–2317, 2014. doi: 10.1002/sec.940.

[3] Z. Guo, M. Li, and X. Fan, "Attribute-based ring signcryption scheme," *Secur. Commun. Netw.*, vol. 6, no. 6, pp. 790–796, 2013. doi: 10.1002/sec.614.

[4] Y. Han, W. Lu, and X. Yang, "Attribute-based signcryption scheme with non-monotonic access structure," in *Proc. INCoS*, Sep. 2013, pp. 796–802. [Online]. Available: http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6630246

[5] H. Hong, X. Liu, and Z. Sun, "A fine-grained attribute based data retrieval with proxy re-encryption scheme for data outsourcing systems," in *Mobile Networks and Applications*. Springer, 2018, pp. 1–6.

[6] Y. Zheng, "Digital signcryption or how to achieve cost (signature & encryption)≪ cost (signature)+ cost (encryption)," in *Proc. Adv. Cryptol.-CRYPTO 17th Annu. Int. Cryptol. Conf.* Lecture Notes in Computer Science, vol. 1294. Santa Barbara, CA, USA: Springer, 1997, pp. 165–179.

[7] H. K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-based signatures," in *CT-RSA*, (Lecture Notes in Computer Science), vol. 6558, A. Kiayias, Ed. Berlin, Germany: Springer, 2011, pp. 376–392. doi: 10.1007/978-3-642-19074-2.

[8] X. Xiang, H. Li, M. Wang, and Z. Liu, "Hidden attribute-based signcryption scheme for lattice," *Secur. Commun. Netw.*, vol. 7, no. 11, pp. 1780–1787, 2014. doi: 10.1002/sec.875.

[9] H. Hong and Z. Sun, "Achieving secure data access control and efficient key updating in mobile multimedia sensor networks," *Multimedia Tools Appl.*, vol. 77, no. 4, pp. 4477–4490, 2018.

[10] H. Hong and Z. Sun, "Sharing your privileges securely: A key-insulated attribute based proxy re-encryption scheme for IoT," *World Wide Web*, vol. 21, no. 3, pp. 595–607, 2018.

[11] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1484–1509, Oct. 1997. doi: 10.1137/S0097539795293172.

[12] J. Proos and C. Zalka, "Shor's discrete logarithm quantum algorithm for elliptic curves," *Quantum Inf. Comput.*, vol. 3, no. 4, pp. 317–344, 2003. [Online]. Available: http://portal.acm.org/citation.cfm?id=2011531

[13] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. STOC*, M. Mitzenmacher, Ed. New York, NY, USA: ACM, 2009, pp. 169–178. doi: 10.1145/1536414.1536440.

[14] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," in *Proc. IEEE 52nd Annu. Symp. Found. Comput. Sci. (FOCS)*. Washington, DC, USA: IEEE Comput. Soc., Oct. 2011, pp. 97–106. doi: 10.1109/FOCS.2011.12.

[15] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(leveled) fully homomorphic encryption without bootstrapping," *ACM Trans. Comput. Theory*, vol. 6, no. 3, p. 13, 2014.

[16] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical GapSVP," in *Advances in Cryptology-CRYPTO* (Lecture Notes in Computer Science), vol. 7417, R. Safavi-Naini and R. Canetti, Eds. Berlin, Germany: Springer, 2012, pp. 868–886. doi: 10.1007/978-3-642-32009-5_50.

[17] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," in *Advances in Cryptology-CRYPTO* (Lecture Notes in Computer Science), vol. 8042, R. Canetti and J. A. Garay, Eds. Berlin, Germany: Berlin, Germany: Springer, 2013, pp. 75–92. doi: 10.1007/978-3-642-40041-4_5.

[18] D.Wichs, "Leveled fully homomorphic signatures from standard lattices," *IACR Cryptol. ePrint Arch.*, vol. 2014, p. 451, 2014. [Online]. Available: http://eprint.iacr.org/2014/451

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

[20] S. Garg, C. Gentry, and S. Halevi, "Candidate multilinear maps from ideal lattices," in *Proc. EUROCRYPT* (Lecture Notes in Computer Science), vol. 7881, T. Johansson and P. Q. Nguyen, Eds. Berlin, Germany: Springer, 2013, pp. 1–17. doi: 10.1007/978-3-642-38348-9.

[21] C. Gentry, S. Gorbunov, and S. Halevi, "Graded multilinear maps from lattices," *IACR Cryptol. ePrint Arch.*, vol. 2014, p. 645, 2014. [Online]. Available: http://eprint.iacr.org/2014/645

[22] X. Boyen, "Attribute-based functional encryption on lattices," in *Proc. TCC*, 2013, pp. 122–142. doi: 10.1007/978-3-642-36594-2_8.

[23] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *Proc. 27th Annu. ACM Symp. Theory Comput. (STOC)*. New York, NY, USA: ACM, 2005, pp. 84–93. doi: 10.1145/1060590.1060603.

[24] F. Li, F. T. B. Muhaya, M. K. Khan, and T. Takagi, *Lattice-Based Signcryption*. Hoboken, NJ, USA: Wiley, 2012. doi: 10.1002/cpe.2826.

[25] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proc. 40th Annu. ACM Symp. Theory Comput. (STOC)*. New York, NY, USA: ACM, 2008, pp. 197–206. doi: 10.1145/1374376.1374407.

[26] S. Agrawal, D. Boneh, and X. Boyen, "Efficient lattice (H)IBE in the standard model," in *Advances in Cryptology-EUROCRYPT* (Lecture Notes in Computer Science), vol. 6110, H. Gilbert, Ed. Berlin, Germany: Springer, 2010, pp. 553–572. 10.1007/978-3-642-13190-5_28.

[27] D. Aharonov and O. Regev, "Lattice problems in NP ∩ coNP," *J. ACM*, vol. 52, no. 5, pp. 749–765, Sep. 2005. doi: 10.1145/1089023.1089025.

[28] D. Micciancio and S. Goldwasser, *Complexity of Lattice Problems—A Cryptographic Perspective*, vol. 671. Springer, 2002.

[29] 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.

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

[31] M. N. Wegman and J. L. Carter, "New hash functions and their use in authentication and set equality," *J. Comput. Syst. Sci.*, vol. 22, no. 3, pp. 265–279, 1981. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0022000081900337

[32] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert, "Bonsai trees, or how to delegate a lattice basis," *J. Cryptol.*, vol. 25, pp. 601–639, Oct. 2012. doi: 10.1007/s00145-011-9105-2.

[33] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Proc. EUROCRYPT* (Lecture Notes in Computer Science), vol. 6632, K. G. Paterson, Ed. Berlin, Germany: Springer, 2011, pp. 568–588. doi: 10.1007/978-3-642-20465-4.

[34] S. Agrawal, X. Boyen, V. Vaikuntanathan, P. Voulgaris, and H. Wee, "Fuzzy identity based encryption from lattices," *IACR Cryptol. ePrint Arch.*, vol. 2011, p. 414, 2011. [Online]. Available: http://eprint.iacr.org/2011/414

[35] D. Cash, D. Hofheinz, and E. Kiltz, "How to delegate a lattice basis," *IACR Cryptol. ePrint Arch.*, vol. 2009, p. 351, 2009. [Online]. Available: http://eprint.iacr.org/2009/351

[36] X. Boyen, "Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more," in *Public Key Cryptography-PKC* (Lecture Notes in Computer Science), vol. 6056, P. Q. Nguyen and D. Pointcheval, Eds. Berlin, Germany: Springer, 2010, pp. 499–517. doi: 10.1007/978-3-642-13013-7_29.

**JIANHUA YAN** received the B.S. degree in chemistry from Jilin University, Changchun, China, in 2002, the M.S. degree in computer science technology from the Liaoning University of Petroleum and Chemical Technology, Fushun, China, in 2005, and the Ph.D. degree in cryptography from the Beijing University of Posts and Telecommunications, Beijing, China, in 2015. He is currently with Ludong University. Meanwhile, he is also an Associate Researcher with the Yantai Research Institute of New Generation Information Technology, Southwest Jiaotong University. His research interests include post quantum cryptography and the Internet security.

**LICHENG WANG** received the B.S. degree in mathematics from Northwest Normal University, Lanzhou, China, in 1995, the M.S. degree in mathematics from Nanjing University, Nanjing, in 2001, and the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2007. He is currently an Associate Professor with the Beijing University of Posts and Telecommunications. His current research interests include modern cryptography, network security, and trust management.

**MUZI LI** received the B.S. degree in mathematics from the Liaoning Institute of Technology, Shenyang, in 1997, and the M.S. degree in computer science technology from the Liaoning University of Petroleum and Chemical Technology, Fushun, China, in 2006. She is currently with Ludong University. Her research interests include information security and the Internet security.

**HASEEB AHMAD** received the B.S. degree from G.C. University, Faisalabad, Pakistan, in 2010, the M.S. degree from the Virtual University of Pakistan, in 2012, and the Ph.D. degree in computers from the Beijing University of Posts and Telecommunications, Beijing, China, in 2017. He is currently with the Department of Computer Science, National Textile University, Faisalabad. His current research interest includes data mining, information retrieval, and information security.

**JUN YUE** received the B.S. degree in mathematics from Ludong university, in 1992, the M.S. degree in computer application technology, in 2003, and the Ph.D. degree in management science and engineering from China Agricultural University, in 2007. She is currently a Professor with the College of Information and Electrical Engineering, Ludong University. Her research interests include big data computing, and the Internet of Things application technologies.

**WENBIN YAO** received the B.S. degree in computer science technology, the M.S. degree in computer architecture, and the Ph.D. degree in computer architecture from the Harbin Institute of Technology, in 1994, 1997, and 2001, respectively. He is currently a Professor with the Beijing University of Posts and Telecommunications, and a member of the Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia. His research interests include disaster recovery, fault-tolerant computing, and trusted computing.