Hindawi
Security and Communication Networks
Volume 2022, Article ID 9919089, 15 pages
https://doi.org/10.1155/2022/9919089

*Research Article*

# A Privacy Preserving Authentication Scheme for Heterogeneous Industrial Internet of Things

**Zuowen Tan, Jintao Jiao, and Mengjiang Yu**

*School of Information Technology, Jiangxi University of Finance and Economics, Nanchang 330032, Jiangxi Province, China*

Correspondence should be addressed to Jintao Jiao; jiaojintao@163.com

Received 14 July 2022; Revised 21 September 2022; Accepted 27 September 2022; Published 17 October 2022

Academic Editor: Kuo-Hui Yeh

Nowadays, the industrial Internet of Things (IIoT) is playing a promising role in the optimization of industrial systems. IIoT devices generate a great amount of data that could be used for different applications. Due to the untrusted nature of machine-to-machine (M2M) communication channels, data authenticity and integrity are an important issue that must be addressed. Especially, it is challenging to deal with the privacy disclosure that the heterogeneity of IIoT brings about. In this paper, we propose a privacy-preserving scheme for authenticity in heterogeneous IIoT systems. Our authentication schemes support many kinds of IIoT devices with multicryptographic configurations such as RSA-based, DL-based, ECC-based, and lattice-based cryptosystems. We provide the formalized proof of the unforgeability and privacy of the proposed schemes in the random oracle model. The experimental simulation demonstrates that the proposed schemes are feasible in the heterogeneous IIoT environment.

## 1. Introduction

With the great increment of smart devices, the Internet of Things (IoT) has made tremendous changes in the way people live. Smart devices in the IoT can connect with each other and exchange information with each other. Among all fields of the IoT, the industrial IoT (IIoT) is the most important one [1]. IIoT has the characteristics of real-time, automation, information interconnection, and so on. By introducing many kinds of sensors, wireless communications, artificial intelligence, and other technologies into the industrial production process, IIoT has greatly improved productivity and quality. Meanwhile, the production costs and resource consumption are reduced, and the intelligence degree of the traditional industry has been promoted to a higher level. Despite its great convenience, one of the major obstacles to widely adopting IIoT is its security risks [2]. Since the IIoT is a network with highly coupled heterogeneous devices, the industries are mostly concerned with the integrity and authenticity of the data which is generated and transmitted by the IIoT devices. In most cases, the IIoT data are transmitted via a public channel. These data are easily intercepted by the adversary. Hence, it is essential to ensure that the data are from legal devices, and the data are not tampered with by a malicious adversary.

Signature schemes are usually used to ensure integrity and authenticity. Nevertheless, since the nature of the existing signature schemes is homogeneous, most of them can only provide data authenticity in a single cryptosystem. Due to the heterogeneity of IIoT, multiple cryptographic systems are always applied for data authenticity [3, 4], and such cryptographic systems are usually based on RSA, DL, and ECC. In addition, due to the consideration of post-quantum security, a cryptographic system based on lattice has already been implemented in IIoT [5]. On the other hand, the privacy of the data sender in IIoT is very important under many circumstances. If the public key is used to verify the validity of the signature, the privacy of the sender's identity will be leaked.

### 1.1. Motivation.

Due to the complexity of practical network architecture in the IIoT environment, it is urgent to design an authentication scheme for the heterogeneous IIoT systems with distinct cryptosystems. Li et al. [6] proposed the SAMA scheme which considers the situation that the devices in IIoT use the same cryptographic configuration. However, it is quite possible that devices in IIoT networks may use different kinds of cryptographic systems. In order to remove this limitation, Wei et al. [7] propose an improved authentication scheme with different system parameters (hereafter, called improved SAMA). However, the improved SAMA scheme only considers the RSA nodes and DL nodes. In fact, there are more devices that adopt other cryptosystems, such as the elliptical cryptosystem. In recent years, more and more attention is paid to postquantum cryptographic algorithms. The threat from quantum computers will emerge in the future, so antiquantum attack cryptographic algorithms have attracted great attention [8]. Some researchers such as Paul and Guerin [9] and Zhang et al. [10] have already conducted a lot of research on postquantum cryptography to secure IIoT. Among the existed postquantum cryptographies, lattice-based cryptography brings the advantage of high-security guarantees and performance efficiency.

In addition, privacy is another major concern in IIoT [11]. To the best of our knowledge, privacy issues in IIoT have not been treated very well in the existing literature, which is a potentially fatal threat. When the IIoT devices transmit data through communication channels, the adversaries could intercept the messages and reveal the identities of the devices.

Motived by the above limitations, we propose novel message authentication schemes with more kinds of nodes compared with the improved SAMA scheme. Our scheme allows the devices in the IIoT to use RSA-based systems, ElGamal-based systems, ECC-based systems, and quantum cryptography. We also consider the privacy preservation of the IIoT devices while the scheme can provide data authenticity. In order to achieve these two goals at the same time, we chose the ring signature to design the authentication schemes in the heterogeneous IIoT environment.

*1.2. Contribution.* We construct novel authentication schemes over a multicryptosystem (hereafter, called ASMC) for the heterogeneous IIoT environment. The main contributions of this paper are as follows:

(1) We propose two ASMC schemes that support heterogeneous devices with multicryptographic configurations such as RSA-based, DL-based, ECC-based, and lattice-based cryptosystems. This solution brings greater flexibility to the authentication for heterogeneous IIoT. Moreover, the ASMC scheme can provide the devices with strong privacy protection.

(2) Based on the hardness assumption of the complex problem, we give the proof of unforgeability against adaptive chosen-message-and-chosen-ring attacks and privacy under the random oracle model.

(3) The IIoT devices only need to do lightweight signing operations online; some expensive operations can be preprocessed offline. So, even if there are lattice-based nodes that may need more computation power, our schemes are still suitable for the IIoT environment.

*1.3. Related Work.* In order to achieve data authenticity [12], researchers have conducted a lot of related works.

In these years, considerable efforts have been paid for the authentication scheme for the resource constrained devices. Bali and Kumar [13] proposed secure clustering for efficient data dissemination among vehicles, and a trust metric was presented by considering various transmission characteristics of vehicles. He et al. [14] proposed a privacy-preserving data aggregation scheme for the smart grid against internal attacks. Bordel et al. [15] applied watermarking and physically unclonable functions to their scheme which can authenticate data in the IoT system, and the scheme is also suitable for 5G networks. Roy et al. [16] proposed a lightweight authentication scheme based on cryptographic hash, bitwise, and fuzzy extractor functions. Peng et al. [17] proposed a mechanism called verifiable query layer (VQL) which can work effectively and guarantee data authenticity. Jain and Prabhakar [18] constructed a system to ensure authenticity and integrity of data in a dynamic database where the server could be untrusted. Challa et al. [19] designed a scheme between a smart meter and a cloud server, and they also give the security and cost analysis of the scheme.

At the same time, the digital signature schemes are usually used to ensure data authenticity for the devices that need to transmit data to each other. Ring signature is one of the important cryptographic primitives which can provide both message authentication and the anonymity of the actual signer. The first ring signature scheme was designed by Rivest et al. [20] in 2001. Any member in the ring can anonymously sign a message, and any receiver can verify the authenticity of the signature. Herranz [21] proposed a ring signature scheme based on RSA. In 2008, Ren and [22] introduced a ring signature scheme based on the ElGamal signature. Recently, some lattice-based ring signature schemes against quantum attacks have been proposed. Liu et al. [23] proposed a lattice-based ring signature that is proven secure against chosen-message attacks. Mundhe et al. [24] proposed a lattice-based ring signature that can provide identity privacy and location privacy. Ren et al. [25] proposed a lattice-based linkable ring signature scheme based on the Borromean ring signature.

*1.4. Paper Organization.* The remaining paper is organized as follows: in Section 2, we present some preliminaries. Section 3 introduces the threat model and security model. Section 4 constructs our ASMC schemes and gives its security analysis. In Section 5, we compare the proposed protocols with the relevant improved SAMA scheme in terms of computation cost and communication overhead. Finally, Section 6 concludes this paper.

## 2. Preliminaries

In this section, we will briefly review some cryptographic assumptions. Table 1 lists the notations used throughout this paper. Preliminaries.

Table 1: The list of notations.

| Notations | Descriptions |
|---|---|
| $k$ | System security parameter |
| $PP$ | Public system parameters |
| $d_i/e_i$ | Private/public key of the $i$th RSA node |
| $N_i$ | Module of the $i$th RSA node |
| $x_i/y_i$ | Private/public key of the $i$th DL node |
| $\mathbb{G}_{DL}$ | Multiplicative cyclic group of prime order $p$ |
| $g$ | Generator of the group $\mathbb{G}_{DL}$ |
| $k_i/K_i$ | Private/public key of the $i$th ECC node |
| $p, q$ | Two great primes |
| $\mathbb{F}_p$ | Finite field of prime order p |
| $\mathbb{G}_{EC}$ | a cyclic group of all points on the elliptic curve EC |
| $G$ | Generator of $\mathbb{G}_{EC}$ |
| $n_{EC}$ | The prime order of $\mathbb{G}_{EC}$ |
| $x_{EC}(\cdot)$ | The $x$-coordinate of an elliptic curve point |
| $\widehat{l}_i/\widehat{L}_i$ | Private/public key of the $i$th lattice node |
| $\mathbb{D}$ | The quotient polynomial ring $\mathbb{Z}_{q_{Lattice}}[x]/(x^{n_{Lattice}}+1)$ |
| $x \overset{\$}{\leftarrow} S$ | is a uniformly random sample drawn from a set $S$ |

*2.1. Elliptic Curve over $\mathbb{F}_p$.* Let $p$ be an odd prime with $p \equiv 3 \bmod 8$. An elliptic curve $EC$ over $\mathbb{F}_p$ is defined by the equation $y^2 = x^3 + ax + b$, where $a, b \in \mathbb{F}_p$, and $\Delta = 4a^3 + 27b^2 0 \equiv \pmod p$. The set $\mathbb{G}_{ec}$ consists of all points $(x, y)$, $x \in \mathbb{F}_p$, $y \in \mathbb{F}_p$, which satisfy the above equation, together with an additional point $\mathcal{O}$ called the point at infinity.

*2.2. Complexity Assumptions.* DLP: Given a finite cyclic group $\mathbb{G}_{dl}$, a generator $g$ of $\mathbb{G}_{dl}$, and an element $h$, the DLP is to find the integer $a$, $0 \le a \le |\mathbb{G}_{dl}| - 1$, such that $h = g^a$.

ECDLP: Given an elliptic curve $EC$ defined over a finite field $\mathbb{F}_p$, a point $P$ of order $n$ in $\mathbb{G}_{ec}$, and a point $Q$ that is a multiple of $P$, the ECDLP is to find the integer $l \in [0, n_{ec} - 1]$, such that $Q = lP$.

SVP (shortest vector problem): Given a monic polynomial $f$ and a lattice $\mathbb{L}$ corresponding to an ideal in the ring $\mathbb{Z}[x]/(f)$, for $\gamma \ge 1$, the $\mathrm{SVP}_\gamma(\mathbb{L})$ is to find an element $g \in \mathbb{L}$ such that $\|g\|_\infty \le \gamma\lambda_1$, where $\lambda_1$ is the shortest length of a nonzero vector in $\mathbb{L}$ [26].

Lyubashevsky and Micciancio [27] introduced a family of collision-resistant hash functions based on the worst-case hardness of standard lattice problems over ideal lattices.

For any integer $z$ and $D_h \subseteq \mathbb{D}$, let $H(\mathbb{D}, D_h, m) = \{h_{\widehat{a}}\colon \widehat{a} \in \mathbb{D}^z\}$ be the family of functions such that for any $\widehat{b} \in D_h^z$, $h_{\widehat{a}}(\widehat{b}) = \widehat{a} \cdot \widehat{b} = \sum_{i \in [z]} a_i b_i$, where $\widehat{a} = (a_1, \ldots, a_z)$, $\widehat{b} = (b_1, \ldots, b_z)$, $[z] = \{1, \ldots, z\}$, and all the operations $a_i b_i$ are performed in the quotient polynomial ring $\mathbb{D}$.

Given an element $h_{\widehat{a}} \in H(\mathbb{D}, D_h, z)$, the collision problem $\mathrm{Col}(h_{\widehat{a}}, D_h)$ of lattice-based hash function is to find distinct elements $\widehat{b}_1$ and $\widehat{b}_2$ in $D_h^z$ such that $h_{\widehat{a}}(\widehat{b}_1) = h_{\widehat{a}}(\widehat{b}_2)$. The function family $H(\mathbb{D}, D_h, m)$ is collision-resistant when the input domain is suitably chosen in $D_h^z \subset \mathbb{D}^z$.

**Theorem 1** (Hardness of collision-resistant hash function [27]). *Let $\mathbb{D}$ be the ring $\mathbb{Z}_{q_{Lattice}}[x]/(x^{n_{Lattice}} + 1)$ for $n_{Lattice}$ a power of two. Define the set $D_h = \{y \in \mathbb{D}\colon \|y\|_\infty \le d\}$ for some integer d. Let $H(\mathbb{D}, D_h, z)$ be a hash function family as the above definition such that $z > log(q_{Lattice})/log(2\ d)$ and $q_{Lattice} \ge 4\ dz\ n_{Lattice}^{1.5} log(n_{Lattice})$. If there is a polynomial-time algorithm that solves $Col(h_{\widehat{a}}, D_h)$ for random $h_{\widehat{a}} \in H(\mathbb{D}, D_h, z)$ with some non-negligible probability, then there is a polynomial-time algorithm that can solve $SVP_\gamma(\mathbb{L})$ for every lattice corresponding to an ideal in $\mathbb{D}$, where $\gamma = 16\ dzn\, log^2(n_{Lattice})$.*

In this paper, we set $d = zn_{\mathrm{Lattice}}^{1.5}\log(n_{\mathrm{Lattice}}) + \sqrt{n_{\mathrm{Lattice}}}\log(n_{\mathrm{Lattice}})$. The setting ensures that the conditions required by the above theorem are verified, and finding collisions for $H(\mathbb{D}, D_h, z)$ implies an algorithm for breaking SVP in the case over ideal lattices for polynomial gaps.

*2.3. Statistical Distance.* Statistical distance is a measure of the difference between two probability distributions [28]. Let $X$ and $X'$ be two random variables over a countable set $S$. The statistical distance between $X$ and $X'$ is defined by

$$\triangle(X, X') = \frac{1}{2}\sum_{x \in S}\left|\Pr[X = x] - \Pr[X' = x]\right|. \quad (1)$$

## 3. System Model

This section gives a brief discussion of the network, threat, and security model of the proposed scheme. We assume that the IIoT is heterogeneous, and the devices may use different cryptographic systems, and a privacy-preserving authentication scheme (ASMC) based on ring signature over a multicryptosystem is proposed under such circumstance.

*3.1. Network Model.* In the proposed ASMC scheme, we consider the network model with cloud data centers for the IIoT device authenticity and privacy. The network model is shown in Figure 1. Specifically, we consider an IIoT network where the devices may be deployed by different workshops or different factories for collecting many types of data. Hence, these devices may use various system parameters. We assume that all the system parameters, public, and private keys have been stored in the devices before they are deployed. After the deployment, devices will collect data and transmit the data to the cloud data centers.

*3.2. Attack Model.* The IIoT devices connect with each other via an insecure channel. During the data transmitting process, an adversary can eavesdrop, intercept, and modify the transmitted data of both devices. So, the IIoT devices are vulnerable to various attacks. We apply the Dolev–Yao (DY)

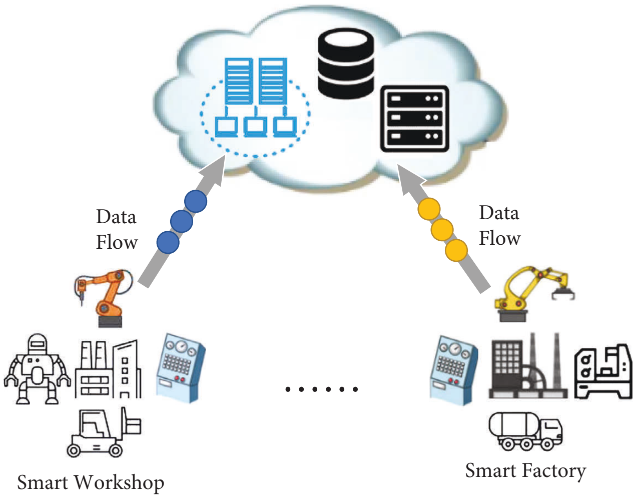

Figure 1: Network model.

attack model [29] in the presented scheme, and there are two types of adversaries in the system.

(1) Passive adversary $\mathscr{A}$ can eavesdrop on the data transmitted in the public channel, such as collecting the messages sent by the IIoT devices. Then, the adversary may analyze the intercepted message and try to reveal the content of the message and identify the real data sender. Since one type of IIoT device always generates a specific kind of data, thus the kind of data implies the type of device. So, the disclosure of the message type may leak the sender's privacy such as identity.

(2) Active adversary $\mathscr{A}$ may actively launch different attacks which include man-in-the-middle, brute force attack, data injection, and so on. $\mathscr{A}$ can intercept and modify the messages during data transmission or inject fake messages into the data channel. An IIoT device may be corrupted and controlled by an active adversary. When an IIoT device is compromised, the adversary can access all the secret information of the device.

### 3.3. Security Model.

In IIoT, many devices transmit data via the insecure channels. In order to mitigate the various types of attacks launched by the passive adversary and active adversary, the ASMC schemes aim at providing the message transmitted with authenticity, integrity, and privacy.

The message authenticity and integrity require that a message is sent by a legal device and has not been altered by any other devices. This security goal is guaranteed by the unforgeability of the underlying ring signature scheme in ASMC. Privacy protection requires that the identity and other private information of the message sender is well protected. The anonymity of the underlying ring signature scheme in ASCM can meet the requirement of privacy. So, the security goals of the ASMC scheme are unforgeability and anonymity.

Unforgeability. The property means that it is difficult for $\mathscr{A}$ to forge a valid signature of an honest ring member in the ASMC scheme. The security model allows the adversary to mount the following two attacks:

(1) Adaptive chosen-message attack: $\mathscr{A}$ can acquire the signature of a message chosen by the adversary in the forge attack phase.

(2) Adaptive chosen-ring attack: $\mathscr{A}$ can choose the ring members and acquire a signature regarding the chosen ring.

In the security model, $\mathscr{A}$ is allowed to make the following oracle queries: *Add*, *Crpt*, *Sign*, and *Hash* (for details, see the proof of Theorem 2). Now, we can define the unforgeability.

*Definition 1.* Existential unforgeability against adaptive chosen-message-and-ring attack (EUF-ACMRA): an ASMC scheme is said to satisfy the existential unforgeability against the adaptive chosen-message-and-ring attack if no probabilistic polynomial-time adversary has a non-negligible advantage Advunforge/ASMC, $\mathscr{A}(k)$ in the experiment Expunforge/ASMC, $\mathscr{A}(k)$ as defined in Table 2, where the advantage of the adversary $\mathscr{A}$ is defined by

$$\mathrm{Adv}_{\mathrm{ASMC},\mathscr{A}}^{\mathrm{unforge}}(k) = \Pr\left[\mathrm{Exp}_{\mathrm{ASMC},\mathscr{A}}^{\mathrm{unforge}}(k) = 1\right]. \tag{2}$$

Anonymity. This property indicates that the ASMC scheme will not reveal the identity of the real data sender.

*Definition 2.* Anonymity

For the ASMC scheme, we define the advantage of any probabilistic polynomial-time adversary $\mathscr{A}$ in the experiment $Exp$anony/$ASMC, \mathscr{A}(k)$ as defined in Table 2, and an ASMC scheme is said to provide the anonymity of the actual signer if the advantage $GI = \left\{(u,v)|SS_{u,v}^{'}|/|SS'| \geq \ 1/q_h(q_h + 1)\right\}$ is negligible for any $\mathscr{A}$ with a security parameter $k$, with querying oracles Add, Reg, Crpt, Sign, $Ch_b$, and Hash (for details, see the proof of Theorem 4).

$$Adv_{ASMC,\mathscr{A}}^{anony}(k) = \left|2\Pr\left[Exp_{ASMC,\mathscr{A}}^{\mathrm{anony}}(k) = b\right] - 1\right|. \tag{3}$$

Table 2: Experiments of unforgeability and anonymity.

| Experiment Expunforge/ASMC, $A(k)$ |
|---|
| Initialization: |
| List←∅; *MList*←∅; *SList*←∅, public parameters $PP$ is generated. |
| Quary1: $\mathscr{A}$ makes oracle queries $Add(\cdot)$, $Reg(\cdot)$, $Crpt(\cdot)$, $Sgin(\cdot)$, and $Hash(\cdot)$ repeatedly. |
| Challenge: $\mathscr{A}$ chooses The message $m'$ and ring $R'$ to challenge. |
| If $pk_i \in R'$ , $(i, sk_i, pk_i) \in$ List, and $i \in MList$, then return 0. |
| Quary2: $\mathscr{A}$ continues to make queries on $Add(\cdot)$, $Reg(\cdot)$, $Crpt(\cdot)$, $Sgin(\cdot)$, and $Hash(\cdot)$ as phase Quary1, but he is not permitted to make query on the oracle $Crpt$ with any $i$ that $pk_i \in R'$ or the oracle $Sign$ with $(R', *, m')$. |
| Forge: $\mathscr{A}$ outputs a forged signature $(R', m', \sigma')$. |
| If Verify$(PP, R', m', \sigma')$ = accpeted, then return 1, else return 0. |
| Experiment $Expanony/ASMC, \mathscr{A}(k)$ |
| **Initialization**: |
| List←∅; *MList*←∅; *SList*←∅, public parameters $PP$ is generated. |
| **Quary1:** $\mathscr{A}$ makes oracle queries $Add(\cdot)$, $Reg(\cdot)$, $Crpt(\cdot)$, $Sgin(\cdot)$, and $Hash(\cdot)$ repeatedly. |
| **Challenge:** $\mathscr{A}$ chooses identities $i_0, i_1$ and a message $m'$, and call oracle $Ch_b(i_0, i_1, m\prime)$. Then, $\mathscr{A}$ obtains a signature $(R', m', \sigma_b)$ where $b \in \{0, 1\}$. |
| **Quary2:** $\mathscr{A}$ continues to make queries on $Add(\cdot)$, $Reg(\cdot)$, $Crpt(\cdot)$, $Sgin(\cdot)$, and $Hash(\cdot)$ as phase Quary1 except the query $Crpt$ with any $i$ that $pk_i \in R'$ or Sign with $(R', *, m')$. |
| **Guess:** $\mathscr{A}$ makes a guess of $b$ denoted by $d$ according to the signature $(R', m', \sigma_b)$. |
| Return the value $d$. |

## 4. 4. Our Authentication Scheme over Multicryptosystem

In the following section, we construct two ASMC schemes, corresponding to two situations, respectively, adjacent nodes using the same cryptographic system as shown in Figure 2 and the mixture of distinct cryptosystem nodes as shown in Figure 3.

*4.1. ASMC for Adjacent Nodes Using the Same Cryptosystem.* In order to provide the legitimacy of one's identity, the signature schemes based on RSA, DL (discrete logarithm), ECC (elliptic curve), and lattice are mostly used at present. For simplicity, we make the assumptions that each node in the IIoT system can use only one type of these four schemes. For adjacent nodes using the same cryptosystem, the ASMC scheme is composed of the following phases.

For clarity, we assume that the number of each type of cryptosystem nodes in the system is $n$. The ring $R$ consists of $n$ RSA nodes (with indices from 1 to $n$), t DL nodes (with indices from $n+1$ to $2n$), $n$ ECC nodes (with indices from $2n+1$ to $3n$), and $n$ lattice nodes (with indices from $3n+1$ to $4n$).

**Setup**$(1^{\mathbf{k}})$. KGC generates $(\mathbb{G}_{DL}, g, p)$ for DL nodes, $(\mathbb{G}_{EC}, G, n_{EC})$ for ECC nodes, and $(\mathbb{D}, S, q)$ for lattice nodes, where $S$ is a nonzero element chosen randomly from $\mathbb{D}$. The parameters $PP$ denote the system public parameters. The hash functions are $H_{DL}\colon \{0,1\}^* \longrightarrow \mathbb{Z}_p$, $H_{EC}\colon \{0,1\}^* \longrightarrow \mathbb{Z}_{n_{EC}}$, and $H_{\text{Lattice}}\colon \{0,1\}^* \longrightarrow \mathbb{Z}_q$.

**KeyGen**(**PP**). After the KeyGen is executed, each node $i$ will be equipped with $(sk_i, pk_i)$. Specially, the key pairs are generated as follows.

Each RSA node $i$ has a private key $sk_i = (d_i, N_i)$ and a public key $pk_i = (e_i, N_i)$. A hash function $H_{RSA}\colon \{0,1\}^* \longrightarrow \mathbb{Z}_{N_{\min}}$ is also generated, where $N_{\min} = \min\{N_1, N_2, \ldots, N_n\}$.

Similarly, each DL node $i$ has a private key $sk_i = x_i$ and a public key $pk_i = y_i$, where $x_i$ is chosen randomly from $[2, p-2]$ and $y_i = g^{x_i}$. Each ECC node $i$ has a private key $sk_i = k_i$ and a public key $pk_i = K_i$, where $k_i \overset{\$}{\leftarrow}$ and $K_i = k_i G$.

For the lattice node $i$, the system generates a vector $(l_1, l_2, \ldots, l_z) \overset{\$}{\leftarrow}$ . If none of the vector component is invertible, then it regenerates the vector. Otherwise, set $\widehat{l}_i = (l_1, l_2, \ldots, l_z)$ as the node's private key. Let $l_j$ be any of the invertible vector components. Then, the system generates $(L_1, L_2, \ldots, L_{j-1}, L_{j+1}, \ldots, L_z) \overset{\$}{\leftarrow}$ and $L_j = l{-}1/j(S - \sum_{k=1,k\neq j}^{z} l_k L_k) \in \mathbb{D}$. The public key is $\widehat{L}_i = (L_1, L_2, \ldots, L_z)$. We have

$$h_{\widehat{L}_i}\left(\widehat{l}_i\right) = \sum_a l_a L_a = S. \tag{4}$$

**RSign**$(\mathbf{PP}, \mathbf{R}, \mathbf{sk_i}, \mathbf{m})$. Suppose that the node $j$ in the ring wants to generate a ring signature. The generation of a ring signature consists of the offline-sign and online-sign phases.

Offline-Sign.

For any RSA node $i \neq j$ in $R$, select $s_i \in \mathbb{Z}_{N_i}^*$ and compute $v_i = s_i^{e_i} \bmod N_i$.

For any DL node $i \neq j$ in $R$, select $s_i \in [2, p-2]$ and compute $v_i = g^{s_i} \bmod p$.

For any ECC node $i \neq j$ in $R$, select $s_i \in \mathbb{Z}_{n_{ec}}^*$ and compute $v_i = s_i G$.

For any lattice node $i \neq j$ in $R$, select $\widehat{s}_i \in \mathbb{D}^z$ and compute $v_i = h_{\widehat{L}_i}(\widehat{s}_i)$.

Online-Sign.

According to the kind of the node, the real signer $j$ generates the ring signature about the message $m$ as shown in the following steps:

(1) Signing by an RSA node

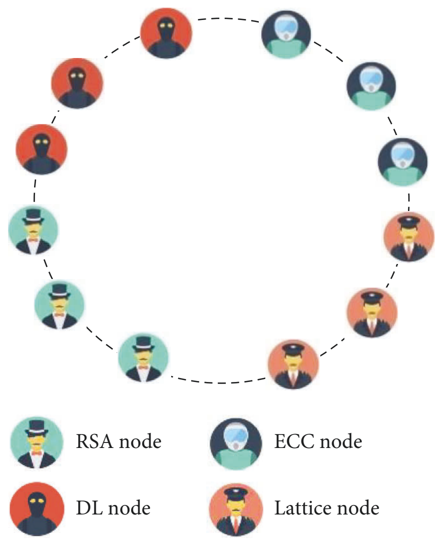

Figure 2: Adjacent nodes using the same cryptographic system.

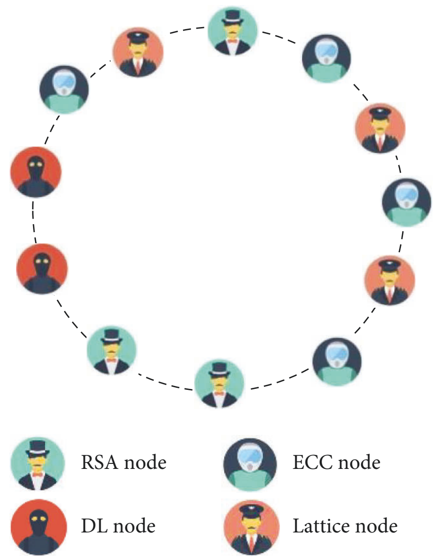

Figure 3: A mixture of distinct cryptosystem nodes.

(1) Node $j$ chooses a random number $\alpha \in \mathbb{Z}_{N_j}^*$ and computes $c_{j+1} = H_{RSA}(R, m, \alpha)$.

(2) For $i = j+1, \ldots, n-1$, Node $i$ computes

$$c_{i+1} = H_{RSA}(R, m, (c_i + v_i) \bmod N_i), \quad (5)$$

(3) Node $n$ computes $c_{t+1} = H_{DL}(R, m, (c_t + v_t) \mathrm{mod} N_n)$.

(4) For $i = n+1, \ldots, 2n-1$, Node $i$ computes

$$c_{i+1} = H_{DL}(R, m, v_i y_i^{c_i} \bmod p). \quad (6)$$

(5) Node $2n$ computes

$$c_{2n+1} = H_{EC}(R, m, v_{2n} y_{2n}^{c_{2n}} \bmod p). \quad (7)$$

(6) For $i = 2n+1, \ldots, 3n-1$, Node $i$ computes

$$c_{i+1} = H_{EC}(R, m, x_{EC}(v_i + c_i K_i) \mathrm{mod}\, n_{EC}). \quad (8)$$

(7) Node $3n$ computes

$$c_{3n+1} = H_{Lattice}(R, m, x_{EC}(v_{3n} + c_{3n} K_{3n}) \bmod n_{EC}). \quad (9)$$

(8) For $i = 3n+1, \ldots, 4n-1$, Node $i$ computes

$$c_{i+1} = H_{Lattice}(R, m, v_i + c_i S). \quad (10)$$

(9) Node $4n$ computes $c_1 = H_{RSA}(R, m, v_{4n} + c_{4n} S)$.

(10) For $i = 1, \ldots, j-1$, Node $i$ computes

$$c_{i+1} = H_{RSA}(R, m, (c_i + v_i) \bmod N_i). \quad (11)$$

(11) Node $j$ computes $s_j = (\alpha - c_j)^{d_j}$.

(2) Signing by a DL node

(1) Node $j$ chooses randomly $\beta \in [2, p-2]$, and Node $i$ computes

$$c_{j+1} = H_{DL}\left(R, m, g^{\beta} \bmod p\right). \tag{12}$$

(2) For $i = j+1, \ldots, 2n-1$, Node $i$ computes

$$c_{i+1} = H_{DL}\left(R, m, v_i y_i^{c_i} \bmod p\right). \tag{13}$$

(3) Node $2n$ computes

$$c_{2n+1} = H_{EC}\left(R, m, v_{2n} y_{2n}^{c_{2n}} \bmod p\right). \tag{14}$$

(4) For $i = 2n+1, \ldots, 3n-1$, Node $i$ computes

$$c_{i+1} = H_{EC}\left(R, m, x_{EC}\left(v_i + c_i K_i\right) \mathrm{mod} n_{EC}\right). \tag{15}$$

(5) Node $3n$ computes

$$c_{3t+1} = H_{\mathrm{Lattice}}\left(R, m, x_{EC}\left(v_{3n} + c_{3n} K_{3n}\right) \bmod n_{EC}\right). \tag{16}$$

(6) For $i = 3n+1, \ldots, 4n-1$, Node $i$ computes

$$c_{i+1} = H_{\mathrm{Lattice}}\left(R, m, v_i + c_i S\right). \tag{17}$$

(7) Node $4n$ computes $c_1 = H_{RSA}\left(R, m, v_{4n} + c_{4n} S\right)$.

(8) For $i = 1, \ldots, n-1$, Node $i$ computes

$$c_{i+1} = H_{RSA}\left(R, m, \left(c_i + v_i\right) \bmod N_i\right). \tag{18}$$

(9) Node $n$ computes

$$c_{n+1} = H_{DL}\left(R, m, \left(c_n + v_n\right) \bmod N_n\right). \tag{19}$$

(10) For $i = n+1, \ldots, j-1$, Node $i$ computes

$$c_{i+1} = H_{DL}\left(R, m, v_i y_i^{c_i} \bmod p\right). \tag{20}$$

(11) Node $j$ computes $s_j = \left(\beta - c_j x_j\right) \mathrm{mod} p$.

(3) Signing by an ECC node

(1) Node $j$ chooses randomly $\gamma \in \mathbb{Z}_{n_{EC}}^{*}$, and it computes

$$c_{j+1} = H_{EC}\left(R, m, x_{EC}(\gamma G) \bmod n_{EC}\right). \tag{21}$$

(2) For $i = j+1, \ldots, 3n-1$, Node $i$ computes

$$c_{i+1} = H_{EC}\left(R, m, x_{EC}\left(v_i + c_i K_i\right) \bmod n_{EC}\right). \tag{22}$$

(3) Node $3n$ computes

$$c_{3n+1} = H_{Lattice}\left(R, m, x_{EC}\left(v_{3n} + c_{3n} K_{3n}\right) \bmod n_{EC}\right). \tag{23}$$

(4) For $i = 3n+1, \ldots, 4n-1$, Node $i$ computes

$$c_{i+1} = H_{Lattice}\left(R, m, v_i + c_i S\right). \tag{24}$$

(5) Node $4n$ computes $c_1 = H_{RSA}\left(R, m, v_{4n} + c_{4n} S\right)$.

(6) For $i = 1, \ldots, n-1$, Node $i$ computes

$$c_{i+1} = H_{RSA}\left(R, m, \left(c_i + v_i\right) \bmod N_i\right). \tag{25}$$

(7) Node $n$ computes

$$c_{n+1} = H_{DL}\left(R, m, \left(c_n + v_n\right) \bmod N_n\right). \tag{26}$$

(8) For $i = n+1, \ldots, 2n-1$, Node $i$ computes

$$c_{i+1} = H_{DL}\left(R, m, v_i y_i^{c_i} \bmod p\right). \tag{27}$$

(9) Node $2n$ computes

$$c_{2n+1} = H_{EC}\left(R, m, v_{2n} y_{2n}^{c_{2n}} \bmod p\right). \tag{28}$$

(10) For $i = 2n, \ldots, j-1$, Node $i$ computes

$$c_{i+1} = H_{EC}\left(R, m, x_{EC}\left(v_i + c_i K_i\right) \bmod n_{EC}\right). \tag{29}$$

(11) Node $j$ computes $s_j = \left(\gamma - c_j k_j\right) \bmod n_{EC}$.

(4) Signing by a lattice node

(1) Node $j$ chooses randomly $\widehat{\delta} \in \mathbb{D}^{z}$, and it computes

$$c_{j+1} = h_{\widehat{L}_j}\left(\widehat{\widehat{\delta}}\right). \tag{30}$$

(2) For $i = j+1, \ldots, 4n-1$, computes

$$c_{i+1} = H_{Lattice}\left(R, m, v_i + c_i S\right). \tag{31}$$

(3) Node $4n$ computes $c_1 = H_{RSA}\left(R, m, v_{4n} + c_{4n} S\right)$.

(4) For $i = 1, \ldots, t-1$, computes

$$c_{i+1} = H_{RSA}\left(R, m, \left(c_i + v_i\right) \bmod N_i\right). \tag{32}$$

(5) Node $n$ computes

$$c_{n+1} = H_{DL}\left(R, m, \left(c_n + v_n\right) \bmod N_n\right). \tag{33}$$

(6) For $i = n+1, \ldots, 2n-1$, Node $i$ computes

$$c_{i+1} = H_{DL}\left(R, m, v_i y_i^{c_i} \bmod p\right). \tag{34}$$

(7) Node $2n$ computes

$$c_{2n+1} = H_{EC}\left(R, m, v_{2n} y_{2n}^{c_{2n}} \bmod p\right). \tag{35}$$

(8) For $i = 2n, \ldots, 3n-1$, Node $i$ computes

$$c_{i+1} = H_{EC}\left(R, m, x_{EC}\left(v_i + c_i K_i\right) \bmod n_{EC}\right). \tag{36}$$

(9) Node $3n$ computes

$$c_{3n+1} = H_{Lattice}\left(R, m, x_{EC}\left(v_{3n} + c_{3n} K_{3n}\right) \bmod n_{EC}\right). \tag{37}$$

(10) For $i = 3n+1, \ldots, j-1$, Node $i$ computes

$$c_{i+1} = H_{Lattice}\left(R, m, v_i + c_i S\right). \tag{38}$$

(11) Node $j$ computes $\widehat{s}_j = \widehat{\delta} - c_j \widehat{l}_j$.

Let $\sigma = \left(c_1, s_1, \ldots, s_{4n}\right)$. Then, the final ring signature on the message $m$ is $(R, m, \sigma)$.

$$\mathbf{Verify}\left(\mathbf{PP}, \mathbf{R}, \mathbf{m}, \boldsymbol{\sigma}\right). \tag{39}$$

Once receiving an ASMC signature $(R, m, \sigma)$, one can check the authenticity and integrity regarding $m$ as follows:

(1) For $i = 1, \ldots, n-1$, the verifier computes

$$c_{i+1} = H_{RSA}\left(R, m, \left(c_i + s_i^{e_i}\right) \bmod N_i\right). \tag{40}$$

(2) The verifier computes $c_{n+1} = H_{DL}\left(R, m, \left(c_n + s_n^{e_n}\right) \bmod N_n\right)$.

(3) For $i = n+1, \ldots, 2n-1$, the verifier computes

$$c_{i+1} = H_{DL}\left(R, m, g^{s_i} y_i^{c_i} \bmod p\right). \tag{41}$$

(4) The verifier computes $c_{2n+1} = H_{EC}\left(R, m, g^{s_{2n}} y_{2n}^{c_{2n}} \bmod p\right)$.

(5) For $i = 2n+1, \ldots, 3n-1$, the verifier computes

$$c_{i+1} = H_{EC}\left(R, m, x_{EC}\left(s_i G + c_i K_i\right) \bmod n_{EC}\right). \tag{42}$$

(6) The verifier computes

$$c_{3n+1} = H_{Lattice}\left(R, m, x_{EC}\left(s_{3n} G + c_{3n} K_{3n}\right) \bmod n_{EC}\right). \tag{43}$$

(7) For $i = 3n+1, \ldots, 4n-1$, the verifier computes

$$c_{i+1} = H_{\text{Lattice}}\left(R, m, h_{\widehat{L}_i}\left(\widehat{s}_i\right) + c_i S\right). \tag{44}$$

(8) The verifier computes $c_1' = H_{RSA}\left(R, m, h_{\widehat{L}_{4n}}\left(\widehat{s}_{4n}\right) + c_{4n} S\right)$.

If $c_1 = c_1'$, the signature $(R, m, \sigma)$ is valid. Otherwise, it is invalid.

*4.2. ASMC for the Mixture of Distinct Cryptosystem Nodes.* In this case, we assume that the ring includes RSA nodes, DL nodes, ECC nodes, and lattice nodes, and they are mixed as shown in Figure 3. Now, the ASMC scheme for a mixture of distinct cryptosystems is different from the ASMC scheme for the first case, especially in the **RSign** and **Verify** phase.

**Setup** $(1^{\mathbf{k}})$ and **KeyGen** (**PP**) are the same as those ones of ASMC in the first case.

$$\mathbf{RSign}\left(\mathbf{PP}, \mathbf{R}, \mathbf{sk_i}, \mathbf{m}\right). \tag{45}$$

The Offline-Sign in the case is the same as the Offline-Sign in the first case. Here, we only give the description of Online-Sign.

Online-Sign.

*Step 1.* Initialization.

If Node $j$ is an RSA, DL, ECC, or lattice node, it randomly chooses $\alpha \in \mathbb{Z}_{N_j}^*$, $\beta \in [2, p-2]$, $\gamma \in \mathbb{Z}_{n_{EC}}^*$ or $\widehat{\delta} \in \mathbb{D}^z$, respectively. Then, Node $j$ computes

$$w_j = \begin{cases} \alpha, & \text{Node } j \text{ is a RSA node} \\ g^{\beta} \bmod p, & \text{Node } j \text{ is a DL node} \\ x_{EC}(\gamma G) \bmod n_{EC}, & \text{Node } j \text{ is an ECC node} \\ h_{\widehat{L}_j}(\widehat{\delta}), & \text{Node } j \text{ is a Lattice node} \end{cases}. \tag{46}$$

Next, Node $j$ computes

$$c_{j+1} = \begin{cases} H_{RSA}\left(R, m, w_j\right), & \text{Node } j+1 \text{ is a RSA node} \\ H_{DL}\left(R, m, w_j\right), & \text{Node } j+1 \text{ is a DL node} \\ H_{EC}\left(R, m, w_j\right), & \text{Node } j+1 \text{ is n ECC node} \\ H_{Lattice}\left(R, m, w_j\right), & \text{Node } j+1 \text{ is a Lattice node} \end{cases}. \tag{47}$$

*Step 2.* Forward the sequence.

For $i = j+1, \ldots, n, 1, \ldots j-1$, Node $i$ computes

$$w_i = \begin{cases} c_i + v_i \bmod N_i, & \text{Node } i \text{ is a RSA node} \\ v_i y_i^{c_i} \bmod p, & \text{Node } i \text{ is a DL node} \\ x_{EC}\left(v_i + c_i K_i\right) \bmod n_{EC}, & \text{Node } i \text{ is an ECC node} \\ v_i + c_i S, & \text{Node } i \text{ is a Lattice node} \end{cases}. \tag{48}$$

Next, it computes $c_{i+1}$ with the same manner as above $c_{j+1}$.

*Step 3.* Form the ring. Finally, Node $j$ computes

$$s_j = \begin{cases} \left(\alpha - c_j\right)^{d_j}, & \text{Node } j \text{ is a RSA node} \\ \left(\beta - c_j x_j\right) \bmod p, & \text{Node } j \text{ is a DL node} \\ \left(\gamma - c_j k_j\right) \bmod n_{EC}, & \text{Node } j \text{ is an ECC node} \\ \widehat{\delta} - c_j \widehat{l}_j, & \text{Node } j \text{ is a Lattice node} \end{cases}. \tag{49}$$

Let $\sigma = (c_1, s_1, \ldots, s_n)$. Then, the ring signature on the message $m$ is $(R, m, \sigma)$.

$$\mathbf{Verify}\left(\mathbf{PP}, \mathbf{R}, \mathbf{m}, \boldsymbol{\sigma}\right). \tag{50}$$

Upon receiving an ASMC signature $(R, m, \sigma)$, the receiver can check the authenticity and integrity regarding $m$ by verifying the signature as follows.

The verifier computes $w_i'$ and $c^{'}_{i+1}$ as the same as $w_i$ and $c_{i+1}$ in the online phase. Finally, it computes

$$c_1' = \begin{cases} H_{RSA}\left(R, m, w_n'\right), & \text{Node1 is a RSA node} \\ H_{DL}\left(R, m, w_n'\right), & \text{Node1 is a DL node} \\ H_{EC}\left(R, m, w_n'\right), & \text{Node1 is an ECC node} \\ H_{Lattice}\left(R, m, w_n'\right), & \text{Node1 is a Lattice node} \end{cases}. \tag{51}$$

If $c_1' = c_1$, the verifier believes the authenticity and integrity of the received message. Otherwise, it outputs $\perp$.

## 5. Security Analysis of Our Scheme

*5.1. Message Authenticity and Integrity Analysis.* The proposed ASMC scheme guarantees the message authenticity and integrity, due to the underlying ring signature being existentially unforgeable against adaptive chosen-message-and-chosen-ring attacks.

Our ring signature scheme is an extension of the 1-out-of-n signature scheme proposed by Abe et al. [30]. The theorems of Abe et al. [30] show that if all the nodes in the ring use the same type of cryptosystems such as the RSA-type, DL-type, or ECC-type, the ring signature scheme is unforgeable.

However, Abe et al. [30] did not consider the lattice-type cryptosystem. In the following section, we will show that the ring signature is existentially unforgeable if all the nodes in the ring are lattice nodes.

**Theorem 2.** *If a $(\tau, \varepsilon, q_s, q_h)$-adversary $\mathscr{A}$ exists in the ASMC scheme for all the nodes in the ring being lattice nodes, a $(\eta, \mu)$-simulator $\mathcal{S}$ can find two vectors $\widehat{s}_i$ and $\widehat{s}_i'$ such that $h_{\widehat{L}_i}(\widehat{l}_i) = h_{\widehat{L}_i}(\widehat{l}_i')$ with the probability at least $\mu$ and the cost time at most $\eta$ for at least one node $i$ in $R$. Here, $\eta < 32q_h^2 + 4/p \cdot \tau$ and $\mu > 9/100$ on the condition that $\epsilon > 8q_h^2/q$ and $q > 2q_h q_s$, where $q$ is the order of the quotient polynomial ring $\mathbb{D}$.*

*Proof.* For simplicity, the hash function $H_{Lattice}(\cdot)$ is written as $H(\cdot)$. Thus, $H(\cdot)$ can be treated as that uses $Q_j = (R_j, m_j, w_j)$ as $j$th query and returns $H(R_j, m_j, w_j)$, where $R_j$ is a set of public keys. The experiment $Exp$unforge/$ASMC, \mathscr{A}(k)$. is carried out as follows.

**Initialization** The simulator $\mathcal{S}$ starts the unforgeable experiment with List$\leftarrow\varnothing$, $M$List$\leftarrow\varnothing$, $SList\leftarrow\varnothing$, and $HList\leftarrow\varnothing$.

**Setup** $\mathcal{S}$ generates $(\mathbb{D}, S, q)$ for lattice nodes, where $S$ is a nonzero element chosen randomly from $\mathbb{D}$.

**Query1** $\mathscr{A}$ makes queries on Add, Reg, Crpt, Sign, and hash oracle $H$ with $\mathcal{S}$ repeatedly. These oracles can make responses as follows.

*Add*$(i)$ If $(i, *, *) \in$ List, return $\perp$, otherwise generate. $\widehat{l}_i = (l_1, l_2, \ldots, l_z) \overset{\$}{\leftarrow}$, add $(i, sk_i, pk_i)$ into *List* and return $pk_i$

Reg$(i, pk_i)$ If $(i, *, *) \in$ List, return $\perp$, otherwise set $pk_i$ as the public key of signer with identity $i$, add $(i, \cdot, pk_i)$ to *List* and identity $i$ to $M$List, finally returns $pk_i$.

Crpt$(i)$ If $(i, sk_i, pk_i) \notin$ List, return $\perp$, otherwise add identity $i$ into $M$List and return $sk_i$.

$H(R, m, w)$ If $(R, m, w, v) \in H$List, return $v$, else uniformly choose $v$ from $\mathbb{Z}_q$, add $(R, m, w, v)$ into $H$List and return $v$.

Sign$(R, k, m)$ If $(R, k, m, \sigma) \in S$List, return $\sigma$, else generate the signature as follows.

For any member $i$ in the ring $R$ including $k$, if $(i, *, *) \notin$ List, return $\perp$. Otherwise, there are two cases.

(1) If $(k, sk_k, *) \in$ List, call the $R$Sign$(R, sk_k, m)$ algorithm to create a signature $\sigma$

(2) If $(k, sk_k, *) \notin$ List, do the following four steps: □

*Step 4.* Choose $c_1 \overset{\$}{\leftarrow}$.

*Step 5.* For $i = 1, \ldots, n$, select $\widehat{s}_i \overset{\$}{\leftarrow}$, compute $w_i = h_{\widehat{L}_i}(\widehat{s}_i) + c_i S$, where $S$ is part of system public parameters $PP$, and then we compute $c_{i+1} = H(R_j, m_j, w_i)$ if$i \neq |R_j|$.

*Step 6.* Assign $c_1$ to the value of $H(R_j, m_j, w_n)$ and add $(R_j, m_j, w_n, c_1)$ into $HList$.

*Step 7.* Form the signature as $\sigma = (c_1, \widehat{s}_1, \ldots, \widehat{s}_n)$.

Finally, we return signature $(R, m, \sigma)$ and add $(R, k, m, \sigma)$ into $SList$.

**Challenge** $\mathscr{A}$ chooses the message $m'$ and ring $R' = \{pk_1', \ldots, pk_n'\}$ that he wants to challenge. For any member $k'$ in the ring $R'$, if $((k', sk_{k'}, pk_{k'}) \in List$ and $k' \in MList)$ or $(R', *, m', *) \in S$List, return $\perp$, otherwise continue to carry on the next phase.

**Query2** $\mathscr{A}$ queries Add, Reg, Crpt, Sign, and hash oracle $H$ the same as the phase, Query1, but he is not allowed to query Crpt with anyone in the ring $R'$ and Sign with $(R', k', m)$.

**Forge** Finally, after those phases above, $\mathscr{A}$ returns a forged signature $\sigma'$ for the ring $R'$.

The oracle Sign$(\cdot)$ fails if inconsistencies of $H(\cdot)$ emerge in Step-3. The probability is at most $q_h/q$ where $q$ is the order of the quotient polynomial ring $\mathbb{D}$. So, Sign$(\cdot)$ can succeed $q_s$ times with probability greater than $(1 - q_h/q)^{q_s} \geq 1 - q_h q_s/q$.

Let $\Theta, \Omega$ be the view given to the signing oracle and $\mathscr{A}$. Let $SS$ be a set of $(\Theta, \Omega, H)$ with which $\mathscr{A}$ is successful in forge. For the success probability of $\mathscr{A}$ restricted by $\Theta, \Omega$, and $H$, we have $\Pr[(\Theta, \Omega, H) \in SS] \geq \epsilon$. Let $(R, m, c_1, s_1, \ldots, s_n)$ be a forged signature that $\mathscr{A}$ outputs. We define $w_i = h_{\widehat{L}_i}(\widehat{s}_i) + c_i S$ and $c_{i+1} = H(R, m, w_i)$ for $i = 1, \ldots, n$. Due to the ideal randomness of $H$, there exist queries $Q_j = (R, m, w_i)$ with probability at least $1 - 1/q$, for $i = 1, \ldots, n$. Let $SS'$ be a subset of $SS$. Then, we have

$$\Pr[(\Theta, \Omega, H) \in SS'] \geq \left(\frac{1 - q_h q_s}{q}\right)\left(\frac{1 - 1}{q}\right)\epsilon. \tag{52}$$

Let $\epsilon^{'} = (1 - q_h q_s/q)(1 - 1/q)\epsilon$. Since the queries form a ring, there exists at least one $k \in \{1, \ldots, n\}$ such that $Q_u = (R, m, w_k)$ and $Q_v = (R, m, w_{k-1})$ with $u < v$. Then, $k$ is between the gap of query order. Let $(u, v)$ be a gap index. Note that $u = v$ happens only if $n = 1$. We will classify $SS'$ by the gap indices. Let $SS^{'}_{u,v}$ be a class which yields gap indices $(u, v)$. Hence, there are at most $(2/q_h) + (1/q_h) = q_h(q_h + 1)/2$ classes. By invoking $\mathscr{A}$ with randomly chosen $(\Theta, \Omega, H)$ at most $t_1 = 1/\epsilon'$ times, $\mathcal{S}$ can find at least one $(\Theta, \Omega, H) \in SS^{'}_{u,v}$ for a gap index $(u, v)$ with probability $1 - \exp(-1) \geq 3/5$.

Let $GI = \left\{(u, v)|SS^{'}_{u,v}|/|SS'| \geq 1/q_h(q_h + 1)\right\}$ and $B = \left\{(\Theta, \Omega, H) \in SS^{'}_{u,v}(u, v) \in GI\right\}$. Then, it holds that $\Pr[B|SS'] \geq 1/2$. Due to the heavy-row lemma [31], $(\Theta, \Omega, H)$ that yields the successful run of $\mathscr{A}$ is in $B$ with probability at least 1/2. We split $H$ as $(H^-, c_k)$ where $H^-$ corresponds to the answers to all queries except for $Q_v$

answered with $c_k$. Due to the heavy-row lemma [31], again, with probability at least 1/2, $(\Theta, \Omega, H^-)$ satisfies $\Pr_{c_k'}[(\Theta, \Omega, H^-, c_k') \in SS_{u,v}^{'}] \geq \epsilon^{'}/2q_h(q_h+1)$. We assume $\epsilon > 8q_h^2/q$ and $q > 2q_hq_s$. It holds that $\epsilon^{'}/2q_h(q_h+1) > 1/q$.

By running $\mathscr{A}$ up to $t_2 = (\epsilon^{'}/2q_h(q_h+1) > 1/q)^{-1}$ times with $(\Theta, \Omega, H^-)$ obtained in the first successful run and randomly chosen $c_k'(\neq c_k)$, then with probability at least 3/5, $\mathscr{S}$ finds at least one $c_k'$ such that $(\Theta, \Omega, H^-, c_k') \in SS_{u,v}^{'}$. Since $Q_u$ happens before $Q_v$, $w_i$ will not change. Therefore, $\mathscr{S}$ gets two distinct tuples $(\widehat{s}_k, c_k, \widehat{l}_k)$ and $(\widehat{s}_k', c_k', \widehat{l}_k)$ that satisfies $h_{\widehat{L}_k}(\widehat{s}_k) + c_kS = h_{\widehat{L}_k}(\widehat{s}_k') + c_k'S$. Then, we will have $h_{\widehat{L}_k}(((\widehat{s}_k - \widehat{s}_k')/(c_k' - c_k)) = S = h_{\widehat{L}_k}(\widehat{l}_k)$. Thus, a collision of the lattice-based hash function has been found. The overall success probability is $\mu > 3/5, 1/2, 1/2, 3/5 = 9/100$, and the number of invocations of $\mathscr{A}$ is

$$t_1 + t_2 < \frac{1}{\epsilon^{'}} + \frac{4q_h(q_h+1)}{\epsilon^{'}} < \frac{4}{\varepsilon} + \frac{4 \cdot 4 \cdot 2q_h^2}{\varepsilon} = \frac{32q_h^2 + 4}{\varepsilon}. \tag{53}$$

By applying the similar technique in [30], with the mixture case of multi-type cryptosystems, we have Theorem 3.

**Theorem 3.** *With regard to the ring, R consists of the public keys of n nodes, if a* $(\tau, \varepsilon, q_s, q_h)$*-adversary* $\mathscr{A}$ *does exist, a* $(\eta, \mu)$*-simulator* $\mathscr{S}$ *is allowed to interact with* $\mathscr{A}$ *who can make queries on the hash oracles up to* $q_h$ *times and the signing oracle up to* $q_s$ *times, and then* $\mathscr{S}$ *can compute* $v^d \bmod N_{min}$ *for* $v \overset{\$}{\leftarrow}$ *with probability greater than* $\mu \geq 3/5$ *and running time* $\eta \leq 4q_h^2/\varepsilon\tau$ *or compute the discrete-logarithm of* $y \in R$ *such that* $y = g^x$ *with probability* $\mu \geq 9/100$ *and running time* $\eta \leq 32q_h^2 + 4/\varepsilon\tau$ *or compute elliptic-curve-discrete-logarithm k of* $K \in R$ *such that* $K = kG$ *with probability* $\mu \geq 9/100$ *and running time* $\eta \leq 32q_h^2 + 4/\varepsilon\tau$ *or compute and finds two vectors* $\widehat{s}$ *and* $\widehat{s}^{'}$ *such that* $h_{\widehat{L}}(\widehat{s}) = h_{\widehat{L}}(\widehat{s}^{'})$ *with probability* $\mu \geq 9/100$ *and running time* $\eta \leq 32q_h^2 + 4/\varepsilon\tau$.

So, the proposed ASMC ring signature with the mixture case of multi-type cryptosystems is existentially unforgeable against adaptive chosen-message and chosen-ring attacks.

By combining the proofs of the theorems in [30] and Theorem 2, we can easily get the proof of Theorem 3. Now, we omit the detailed proof.

2. Privacy analysis. The privacy of the proposed ASMC schemes can be guaranteed by the anonymity of the ring signature scheme.

**Theorem 4 (Anonymity).** For $b \in \{0, 1\}$, let $X_{b,R,sk_{i_b},m}$ be the signature of a member $i_b$ in the ring $R$ about message $m$ in the proposed ASMC schemes. With a probabilistic polynomial-time turning machine, the statistical distance (as defined in Eq. 1) between the two signatures generated by signer $i_0$ and signer $i_1$ holds $\Delta(X_{0,R,sk_{i_0},m}, X_{1,R,sk_{i_1},m}) \leq 1/g(k)$ for any polynomial $g$ and sufficiently large $k$ under the random oracle model.

*Proof.* Here, we will prove that the statistical distance between two signatures $X_{0,PP,R,sk_{i_0},m}, X_{1,PP,R,sk_{i_1},m}$ is negligible. There is a simulator $\mathscr{S}$ and an adversary $\mathscr{A}$ in $\mathrm{Exp}_{ASMC,\mathscr{A}}^{\mathrm{anony}-b}(k)$. Assume that $\mathscr{A}$ is granted the ability to access the security key of any member in the ring, and $\mathscr{A}$ is allowed to query $\mathscr{S}$ with the oracles Add, Reg, Crpt, Sign, $Ch_b$, and hash oracles. The anonymity experiment $\mathrm{Exp}_{\mathrm{MEMA},\mathscr{A}}^{\mathrm{anony}-b}(k)$ is carried out asfollows.

**Initialization** The anonymity experiment starts with List$\leftarrow\varnothing$, $M$List$\leftarrow\varnothing$, $S$List$\leftarrow\varnothing H$List$_{RSA}\leftarrow\varnothing$, $H$List$_{DL}\leftarrow\varnothing$, $H$List$_{EC}\leftarrow\varnothing$, and $H$List$_{\mathrm{Lattice}}\leftarrow\varnothing$.

**Setup** $\mathscr{S}$ generates $(\mathbb{G}_{dl}, g, p)$ for DL nodes, $(\mathbb{G}_{EC}, G, n_{EC})$ for ECC nodes, and $(\mathbb{D}, S, q)$ for lattice nodes, where $S$ is a nonzero element chosen randomly from $\mathbb{D}$. The generated parameters are considered as public parameters $PP$.

**Query 1** $\mathscr{A}$ makes queries on Add, Reg, Crpt, Sign, and hash oracles repeatedly. $\mathscr{S}$ makes responses about the oracle queries as follows.

Add$(i)$: if $(i, *, *) \in$ List, return $\perp$, otherwise generate $(sk_i, pk_i)$, add $(i, sk_i, pk_i)$ into List, and return $pk_i$.

Hash$_{\mathrm{RSA}}(m)$: if $(m, v) \in H$List$_{RSA}$, return $v$, else uniformly choose $v$ from $\mathbb{Z}_{N_{min}}$, add $(m, v)$ into $H$List$_{\mathrm{RSA}}$, and return $v$.

Hash$_{DL}(m)$, Hash$_{EC}(m)$, and Hash$_{\mathrm{Lattice}}(m)$ are handled similarly as Hash$_{\mathrm{RSA}}(m)$. $v$ is chosen randomly from the legal domain, and the $(m, v)$ pair is added into the corresponding oracle list $H$List$_{DL}$, $H$List$_{EC}$, or $H$List$_{\mathrm{Lattice}}$.

Sign$(R, k, m)$: If $(R, k, m, \sigma) \in SList$, return $\sigma$, else generate the signature as follows.

For any member $i$ in the ring $R$ including $k$, if $(i, *, *) \notin$ List, return $\perp$. Otherwise, there are two cases.

(1) If $(k, sk_k, *) \in$ List, call the $R$Sign algorithm to create a signature $\sigma$.

(2) If Node 1 is an RSA, DL, ECC, or lattice node and $(k, sk_k, *) \notin$ List, it randomly chooses $c_1$ from $\mathbb{Z}_{N_{\min}}$, $\mathbb{Z}_p$, $\mathbb{Z}_{n_{EC}}$, or $\mathbb{Z}_q$, respectively, and generate $s_i$, $v_i$, $w_i$, $c_{i+1}$ with the same manner as the ASMC scheme.

Lastly, program hash oracle as

$$\begin{cases} \mathrm{Hash}_{RSA}(R, m, w_n) = c_1, & \text{if Node 1 is a RSA node} \\ \mathrm{Hash}_{DL}(R, m, w_n) = c_1, & \text{if Node 1 is a DL node} \\ \mathrm{Hash}_{EC}(R, m, w_n) = c_1, & \text{if Node 1 is an ECC node} \\ \mathrm{Hash}_{\mathrm{Lattice}}(R, m, w_n) = c_1, & \text{if Node 1 is a Lattice node} \end{cases}. \tag{54}$$

Form the signature as $\sigma = (c_1, s_1, \ldots, s_n)$ and add $(R, k, m, \sigma)$ into $S$List.

Finally, return the signature. $(R, m, \sigma)$, Reg$(\cdot)$ and Crpt$(\cdot)$ are handled the same as in Theorem 2.

**Challenge** $\mathscr{A}$ chooses two identities $i_0, i_1$, and message $m'$, the $Ch_b$ oracle is handled as follows.

-$Ch_b(i_0, i_1, m')$ If $(i_0, sk_{i_0}, pk_{i_0})$ or $(i_1, sk_{i_1}, pk_{i_1}) \notin$ List, return $\perp$. Otherwise, we generate a ring $R'$ which contains the public keys of $i_0$ and the public key of $i_1$. If $(R', i_b, m', *) \in S$List, return $\perp$. Otherwise, we generate a signature by calling $\mathrm{Sign}(R', i_b, m')$ where $b \in \{0, 1\}$. Finally return the signature $(R', m', \sigma_b)$. Note that the triple $(R', i_b, m', \sigma_b)$ is also added to the list SList.

**Query 2** The adversary $\mathscr{A}$ queries Add, Reg, Crpt, Sign, and hash oracles in the same way as in Query 1, but he is not allowed to query about Crpt with any $k$ in the ring $R'$ or Sign with $(R', m')$.

Finally, $\mathscr{A}$ returns a guess of bit $b$ according to the signature $(R', m', \sigma_b)$ from $Ch_b(i_0, i_1, m\prime)$.

Since the pair $(R', m')$ of the two signatures $(R', m', \sigma_0)$ and $(R', m', \sigma_1)$ are the same as each other, we will omit it when we measure the statistical distance between $X_{R',m',\sigma_0}$ and $X_{R',m',\sigma_1}$. Set the $\sigma = (c_1, s_1, \ldots, s_n)$ as a vector of $n + 1$ coordinates. Thus, each coordinate is independent of each other. For clarity, $\sigma_0$ and $\sigma_1$ are written as $(c_1', s_1', \ldots, s_{i_0}', \ldots, s_{i_1}', \ldots, s_n')$ and $(c_1'', s_1'', \ldots, s_{i_0}'', \ldots, s_{i_1}'', \ldots, s_n'')$. All the signature coordinates except for $s_{i_0}'$ of $i_0$ are directly chosen uniformly from the domains defined in the ASMC scheme (here, we refer the domain as domain $A$, but the meaning of domain $A$ for distinct kinds of nodes is different), while $s_{i_1}''$ of signature $i_1$ is not directly chosen from the domain $A$. We set $\lambda = \min(N_{\min}, q, n_{EC}, p)$, $\lambda' = \max(N_{\max}, q, n_{EC}, p)$. The distance between $X_{R',m',\sigma_0}$ and $X_{R',m',\sigma_1}$ is

$$\begin{aligned}
\triangle\left(X_{R',m',\sigma_0}, X_{R',m',\sigma_1}\right) &= \sum_{c_1, s_i \in A, i \in \{1,\ldots,n\}} \begin{matrix} \left[\Pr\left[X_{R',m',\sigma_0} = \left(c_1, s_1, \ldots, s_{i_0}, \ldots, s_{i_1}, \ldots, s_n\right)\right]\right] \\ -Pr\left[X_{R',m',\sigma_1} = \left(c_1, s_1, \ldots, s_{i_0}, \ldots, s_{i_1}, \ldots, s_n\right)\right] \end{matrix} \\
&\qquad \Pr(c_1), \Pr\left(s_1, \ldots, s_{i_0-1}, s_{i_0+1}, \ldots, s_{i_1-1}, s_{i_1+1}, \ldots, s_n\right) \\
&= \sum_{c_1, s_i \in A, i \in \{1,\ldots,n\}} \begin{matrix} \left[|\Pr\left(s_{i_0} | s_{i_0-1}\right), \Pr\left(s_{i_1}\right)\right) \\ -Pr\left(s_{i_0}\right), Pr\left(s_{i_1} | s_{i_1-1}\right)|\right] \end{matrix} \\
&\qquad \Pr(c_1), \prod_{i=1}^{t} \Pr(s_i) \\
&= \sum_{c_1, s_i \in A, i \in \{1,\ldots,n\}} \begin{matrix} i \neq i_0, i_1 \\ , \left[|\Pr\left(s_{i_0} | s_{i_0-1}\right), \Pr\left(s_{i_1}\right)\right. \\ \left.-Pr\left(s_{i_0}\right), Pr\left(s_{i_1} | s_{i_1-1}\right)|\right] \end{matrix} \leq \left(\frac{\lambda'}{\lambda}\right)^{t-1} \left(\frac{\lambda'^2}{\lambda^2} - \frac{\lambda^2}{\lambda'^2}\right) \\
&= \left(\frac{\lambda'}{\lambda}\right)^{n-1} \left(\frac{\lambda'}{\lambda} + \frac{\lambda}{\lambda'}\right) \left(\frac{\lambda'^2 - \lambda^2}{\lambda\lambda'}\right) \leq 2\left(\frac{\lambda'}{\lambda}\right)^{t} \left[\left(\frac{\lambda'}{\lambda}\right)^2 - 1\right] \\
&= 2\left(\frac{\lambda'}{\lambda}\right)^{t} \left(\frac{\lambda'}{\lambda} + 1\right) \left(\frac{\lambda'}{\lambda} - 1\right) \\
&\leq 4\left(\frac{\lambda'}{\lambda}\right)^{n+1} \left(\frac{\lambda'}{\lambda} - 1\right),
\end{aligned} \tag{55}$$

The variables $\alpha, \beta, \gamma, \widehat{\delta}$ are used in the real signer's secure value $s_{i_b}$ computation, and they are generated uniformly as $\alpha \overset{\$}{\leftarrow}, \beta \overset{\$}{\leftarrow}, \gamma \overset{\$}{\leftarrow}$, and $\widehat{\delta} \overset{\$}{\leftarrow}$. So, $s_{i_b}$ can still be considered a uniform chosen variable. $c_1$ is generated by $\mathrm{Hash}_{RSA}, \mathrm{Hash}_{DL}, \mathrm{Hash}_{EC}, \mathrm{Hash}_{\mathrm{Lattice}}$ random oracles with uniform distributions as $\mathrm{Hash}_{RSA}\colon \{0,1\}^* \longrightarrow \mathbb{Z}_{N_{\min}}$, $\mathrm{Hash}_{DL}\colon \{0,1\}^* \longrightarrow \mathbb{Z}_p$, $\mathrm{Hash}_{EC}\colon \{0,1\}^* \longrightarrow \mathbb{Z}_n$, $\mathrm{Hash}_{\mathrm{Lattice}}\colon \{0,1\}^* \longrightarrow \mathbb{Z}_q$. So, we have Equation (2).

If the security parameter $k$ is larger enough, $(\lambda'/\lambda - 1) \leq g'(k)$ for any polynomial $g'$, and $4(\lambda'/\lambda)^{n+1}$ will have an upper bound. Then, we have $\triangle(X_{R',m',\sigma_0}, X_{R',m',\sigma_1}) \leq g(k)$ for any polynomial $g$ and sufficiently large $k$. □

## 6. Efficiency Analysis and Experimental Results

In this section, we will give an analysis of the computation cost and the communication overhead of the proposed schemes. To the best of our knowledge, there is no other

Table 3: The comparison of computational cost.

| Scheme | Offline-sign | Online-sign | Verification |
|---|---|---|---|
| Wei et al. [7] (RSA signer) | $n/2\ (S_R + S_D) - S_R$ | $S_R + n/2\, S_D$ | $n/2\ (V_R + V_D)$ |
| Wei et al. [7] (DL signer) | $n/2\ (S_R + S_D) - S_D$ | $n/2\, S_D$ | $n/2\ (V_R + V_D)$ |
| Our scheme (RSA signer) | $n/4 \begin{pmatrix} S_R + S_D \\ +S_E + S_L \end{pmatrix} - S_R$ | $S_R + n/4\ (S_D + S_E)$ | $n/4\ (S_R + S_D + S_E + S_L)$ |
| Our scheme (DL signer) | $n/4 \begin{pmatrix} S_R + S_D \\ +S_E + S_L \end{pmatrix} - S_D$ | $n/4\ (S_D + S_E)$ | $n/4\ (S_R + S_D + S_E + S_L)$ |
| Our scheme (ECC signer) | $n/4 \begin{pmatrix} S_R + S_D \\ +S_E + S_L \end{pmatrix} - S_E$ | $n/4\ (S_D + S_E)$ | $n/4\ (S_R + S_D + S_E + S_L)$ |
| Our scheme (lattice signer) | $n/4 \begin{pmatrix} S_R + S_D \\ +S_E + S_L \end{pmatrix} - S_L$ | $n/4\ (S_D + S_E)$ | $n/4\ (S_R + S_D + S_E + S_L)$ |

Table 4: The comparison of communication overhead.

| Scheme | Communication overhead | No. of bits ($n = 20$) |
|---|---|---|
| Wei et al. [7] | $n/2\,(\lvert\mathbb{G}_{\mathrm{RSA}}\rvert + \lvert\mathbb{G}_{DL}\rvert) + \lvert\mathbb{G}_{\mathrm{RSA}}\rvert$ | 12,864 |
| Our scheme | $n/4\,(\lvert\mathbb{G}_{\mathrm{RSA}}\rvert + \lvert\mathbb{G}_{DL}\rvert + \lvert\mathbb{G}_{EC}\rvert + z\lvert\mathbb{G}\rvert) + \lvert\mathbb{G}_{\mathrm{RSA}}\rvert$ | 339,744 |

Table 5: The computation time (msec.).

| Phase | Value of n | Real signer type | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | RSA | | DL | | ECC | | Lattice | |
| | | Wei et al. [7] | Our scheme | Wei et al. [7] | Our scheme | Wei et al. [7] | Our scheme | Wei et al. [7] | Our scheme |
| Offline sign | 20 | 41.52 | 32.27 | 35.46 | 28.38 | - | 30.23 | - | 30.51 |
| | 40 | 76.65 | 60.07 | 65.45 | 56.07 | - | 55.74 | - | 51.92 |
| | 60 | 119.09 | 93.7 | 100.87 | 84.18 | - | 81.55 | - | 79.09 |
| | 80 | 141.44 | 121.98 | 141.76 | 102.86 | - | 104.46 | - | 108.53 |
| | 100 | 195.26 | 158.78 | 156.01 | 129.93 | - | 129.53 | - | 134.18 |
| | 200 | 358.72 | 310.94 | 348.34 | 245.6 | - | 290.21 | - | 262.25 |
| Online sign | 20 | 31.65 | 23.15 | 23.72 | 20.42 | - | 21.63 | - | 22.45 |
| | 40 | 58.51 | 44.87 | 46.5 | 40.82 | - | 43.18 | - | 43.18 |
| | 60 | 89.96 | 65.74 | 68.84 | 52.39 | - | 56.91 | - | 63.6 |
| | 80 | 145.03 | 106.08 | 111.11 | 99.56 | - | 106.99 | - | 109.41 |
| | 100 | 181.17 | 138.96 | 174.66 | 130.69 | - | 140.93 | - | 144.23 |
| | 200 | 284.59 | 207.31 | 228.58 | 194.36 | - | 203.27 | - | 205.16 |
| Verify | 20 | 113.84 | 87.47 | DL | 76.4 | - | 82.21 | - | 89.17 |
| | 40 | 140.99 | 87.57 | 104.45 | 80.86 | - | 76.15 | - | 91.83 |
| | 60 | 205.48 | 132.88 | 109.05 | 128.2 | - | 121.5 | - | 132.59 |
| | 80 | 331.99 | 224.39 | 168.86 | 207.82 | - | 197.29 | - | 220.03 |
| | 100 | 389.37 | 319.81 | 251.03 | 269.04 | - | 289.02 | - | 299.06 |
| | 200 | 687.8 | 492.57 | 325.09 | 400.02 | - | 390.87 | - | 433.24 |

heterogeneous IIoT authentication system in the literature that supports multiple cryptosystems except for Sun's authentication scheme which supports two different cryptosystems. So, we just compare the scheme proposed by Wei et al. [7] with our schemes.

Table 3 gives the comparison of the computation costs of the proposed ASMC schemes and the scheme proposed by Wei et al. [7]. For the sake of fairness, we assume that the number of nodes in the two schemes is the same. Each ring has $n$ nodes. Furthermore, we assume that the proportion of each type of node in the two schemes is the same. Specifically, there are $n/2$ RSA nodes and $n/2$ DL nodes in [7], while there are $n/4$ RSA nodes, $n/4$ DL nodes, $n/4$ ECC nodes, and $n/4$ Lattice nodes in our scheme.

Let $S_R$ and $V_R$ denote the signature generation and verification exponentiation operations carried out by an RSA node. Similarly, let $S_D$ and $V_D$ denote the signing and verification exponentiation operations carried out by a DL node. Let $S_E$ and $V_E$ represent the signing and verification elliptic curve multiplication operations carried out by an

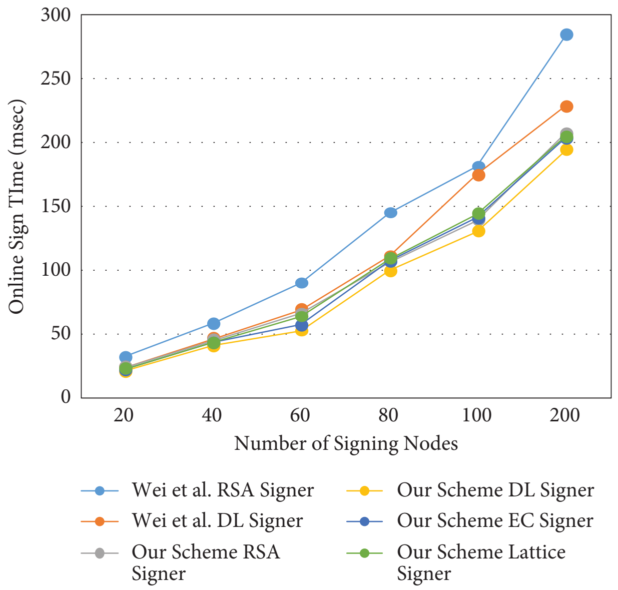

Figure 4: Computational cost for Online-Sign.

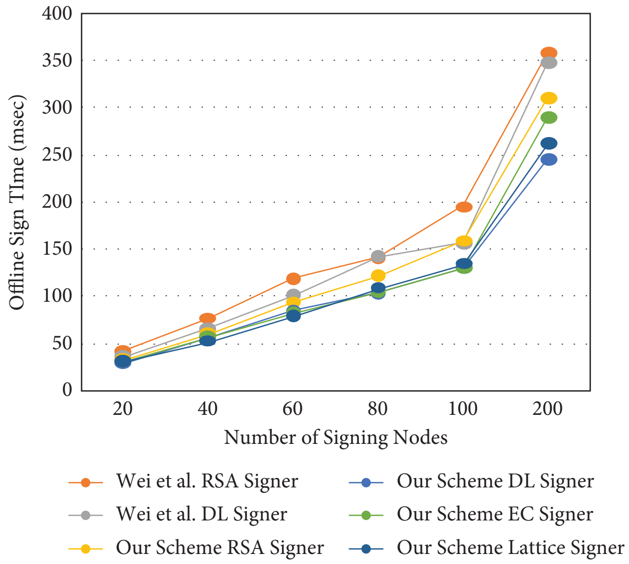

Figure 5: Computational cost for Offline-Sign.

ECC node. Let $S_L$ and $V_L$ represent the signing and verification polynomial multiplication operations carried out by a lattice node.

Table 4 gives the comparison of the communication overhead between Wei et al. [7] and our ASMC scheme. $|\mathbb{G}_{\mathrm{RSA}}|$ is the size of a random number in the signature chosen by the RSA node, while $|\mathbb{G}_{DL}|$ and $|\mathbb{G}_{EC}|$ are the size of random numbers chosen by the DL node and ECC node, respectively. $|\mathbb{G}|$ is the size of the generator of the quotient polynomial ring $\mathbb{D}$, and $z$ is the number of polynomials contained in one vector which is chosen by the lattice node. We set the security level as 1024-bit RSA, 1024-bit DL, 160-bit ECC. As for lattice nodes, we choose $n_{\mathrm{lattice}} = 256$, $z = 16$ and set $q_{\mathrm{lattice}}$ to a 16bit prime number, and we note that the meaning of those symbols is the same as in Theorem 1. As the size of $z|\mathbb{G}|$ is much bigger than the others, so the communication overhead of our ASMC scheme is higher than the scheme of Wei et al. [7], as shown in Table 4. Considering that the communication overhead of lattice-based signatures is generally larger than that of other type signatures, even the communication overhead of our ASMC scheme is several times of Wei et al. [7], it is still acceptable compared with other lattice-based ring signature schemes. When the number of signing nodes is 20, the communication overhead of our ASMC scheme is 42.5 K bytes, while the communication overhead of other lattice-based ring signature schemes is basically of the same order of magnitude as ours or even

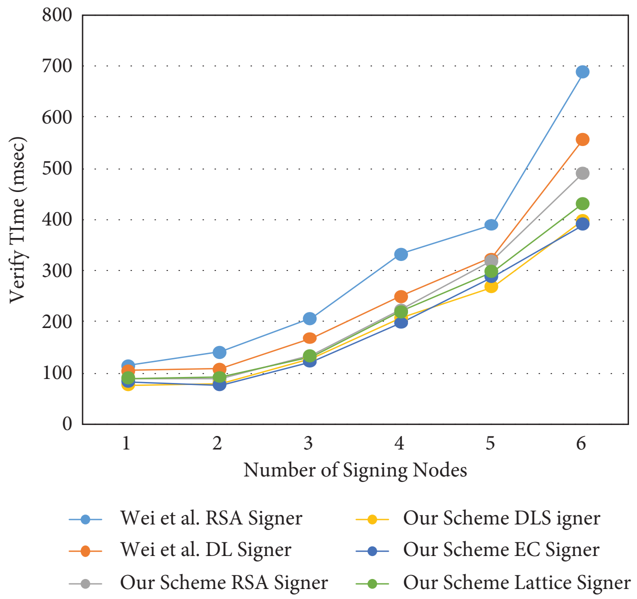

Figure 6: Computational cost for verification.

larger. For instance, the communication overhead of SALRS proposed by Liu et al. [32] is 53.6k with 16 ring members.

Table 5 gives the simulation experiments' computation time based on different values of *n* on a PC. The configuration is Intel Core i7-8550 CPU@2.00 GHz and 16 GB RAM with Window 10 operating system. The signature is generated by using the NTL [33]. The computational time of every operation is calculated as the average time of 10 executions. Our simulation is based on the architecture of adjacent nodes using the same cryptographic system. Since the actual operations remain the same in the architecture of a mixture of distinct cryptosystem nodes, so the results of the two architectures shown in Figures 2 and 3 are similar.

Figures 4–6 show the computation time of a different number of signing nodes in the phase Offline-Sign, Online-Sign, and verification, respectively. From the experiment results, we can see that the computational cost of our ASMC scheme is lower than that of [7], and the computational cost difference between the two schemes increases with the number of signing nodes.

## 7. Conclusion

In this paper, we proposed two novel privacy-preserving message Figure 6 authentication schemes that allow IIoT devices to use different security systems and parameters. Our AMSC schemes can support four kinds of nodes. Analysis shows that the ASMC schemes can provide the authentication and integrity of the message and the anonymity of the message senders.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## Acknowledgments

This work was partially supported by the National Natural Science Foundation of China under Grant no. 61862028, Graduate Innovation Foundation of Jiangxi Province No. YC2020-B123, and the Foundation of Wuyi University under Grant no. 2018L3013.

## References

[1] K. W. Rehman, H. M. Zangoti, M. K. Afzal, N. Armi, and K. Salah, "Industrial Internet of things: Industrial internet of things: Recent advances, enabling technologies and open challengesecent advances, enabling technologies and open challenges," *Computers & Electrical Engineering*, vol. 81, Article ID 106522, 2020.

[2] A. Kaci and A. Rachedi, "Toward a Machine Learning and Software Defined Network Approaches to Manage Miners' Reputation in Blockchain," *Journal of Network and Systems Management*, vol. 28, no. 3, pp. 478–501, 2020.

[3] Q. Jing, A. V. Vasilakos, J. Wan, J. Lu, and D. Qiu, "Security of the internet of things: perspectives and challenges," *Wireless Networks*, vol. 20, no. 8, pp. 2481–2501, 2014.

[4] F. Li, H. Zhang, and T. Takagi, "Efficient Efficient Signcryption for Heterogeneous Systemsigncryption for heterogeneous systems," *IEEE Systems Journal*, vol. 7, no. 3, pp. 420–429, Sept. 2013.

[5] R. V. Begum, N. Shanmugasundaram, and N. Ganesh, "Lattice-based cryptography using internet of things," *International Journal of Information and Computing Science*, vol. 6, no. 4, pp. 444–451, April 2019.

[6] J. Li, Y. Li, J. Ren, and J. Wu, "Hop-by-Hop Hop-by-Hop Message Authenticationand Source Privacy in WirelessSensor Networksessage authentication and source privacy in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 5, pp. 1223–1232, May 2014.

[7] J. Wei, T. V. X. Phuong, and G. Yang, "An An Efficient Privacy Preserving Message Authentication Scheme for Internet-of-Thingsfficient privacy-preserving message authentication scheme for internet-of-things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 1, pp. 617–626, Jan. 2021.

[8] L. Wang, C. Huang, and H. Cheng, "Quantum attack-resistant signature scheme from lattice cryptography for WFH," in *Proceedings of the 2021 IEEE 2nd International Conference on Big Data Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, pp. 868–871, Nanchang, China, March 2021.

[9] S. Paul and E. Guerin, "Hybrid OPC UA: enabling post-quantum security for the industrial internet of things," in *Proceedings of the 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 238–245, Vienna, Austria, September 2020.

[10] X. Zhang, C. Xu, H. Wang, Y. Zhang, and S. Wang, "Lattice-based forward secure public-key encryption with keyword search for cloud-assisted industrial internet of things," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1019–1032, 2021.

[11] T. Wang, M. Z. A. Bhuiyan, G. Wang, L. Qi, J. Wu, and T. Hayajneh, "Preserving Preserving Balance Between Privacy and Data Integrity in Edge-Assisted Internet of Thingsalance between privacy and data integrity in edge-assisted internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2679–2689, 2020.

[12] L. Zhou, K. H. Yeh, G. Hancke, Z. Liu, and C. Su, "Security and Security and Privacy for the Industrial Internet of Things: An Overview of Approaches to Safeguarding Endpointsrivacy for the industrial internet of things: an overview of approaches to safeguarding endpoints," *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 76–87, 2018.

[13] R. S. Bali and N. Kumar, "Secure clustering for efficient data dissemination in vehicular cyber–physical systems," *Future Generation Computer Systems*, vol. 56, pp. 476–492, 2016.

[14] D. He, N. Kumar, and J.-H. Lee, "Privacy-preserving data aggregation scheme against internal attackers in smart grids," *Wireless Networks*, vol. 22, no. 2, pp. 491–502, 2016.

[15] B. Bordel, R. Alcarria, T. Robles, and M. S. Iglesias, "Data Data Authentication and Anonymization in IoT Scenarios and Future 5G Networks Using Chaotic Digital Watermarkinguthentication and anonymization in IoT scenarios and future 5G networks using chaotic digital watermarking," *IEEE Access*, vol. 9, pp. 22378–22398, 2021.

[16] S. Roy, S. Chatterjee, A. K. Das, S. Chattopadhyay, N. Kumar, and A. V. Vasilakos, "On the On the Design of Provably Secure Lightweight Remote User Authentication Scheme for Mobile Cloud Computing Servicesesign of provably secure lightweight remote user authentication scheme for mobile cloud computing services," *IEEE Access*, vol. 5, pp. 25808–25825, 2017.

[17] Z. Peng, H. Wu, B. Xiao, and S. Guo, "VQL: providing query efficiency and data authenticity in blockchain systems," in *Proceedings of the 2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW)*, pp. 1–6, Macao, China, April 2019.

[18] R. Jain and S. Prabhakar, "Guaranteed Authenticity and Integrity of Data from Untrusted Servers," in *Proceedings of the 2014 IEEE 30th International Conference on Data Engineering*, pp. 1282–1285, Chicago, IL, USA, April 2014.

[19] S. Challa, A. K. Das, P. Gope, N. Kumar, F. Wu, and A. V. Vasilakos, "Design and analysis of authenticated key agreement scheme in cloud-assisted cyber–physical systems," *Future Generation Computer Systems*, vol. 108, pp. 1267–1286, 2020.

[20] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*, pp. 552–565, Australia, December2001.

[21] J. Herranz, "Identity-Identity-based ring signatures from RSAased ring signatures from RSA," *Theoretical Computer Science*, vol. 389, no. 1-2, pp. 100–117, 2007.

[22] J. Ren and L. Harn, "Generalized Generalized Ring Signaturesing signatures," *IEEE Transactions on Dependable and Secure Computing*, vol. 5, no. 3, pp. 155–163, 2008.

[23] J. Liu, Y. Yu, J. Jia et al., "Lattice-based double-authentication-preventing ring signature for security and privacy in vehicular Ad-Hoc networks," *Tsinghua Science and Technology*, vol. 24, no. 5, pp. 575–584, 2019.

[24] P. Mundhe, V. K. Yadav, S. Verma, and S. Venkatesan, "Efficient Efficient Lattice-Based Ring Signature for Message Authentication in VANETsattice-based ring signature for message authentication in VANETs," *IEEE Systems Journal*, vol. 14, no. 4, pp. 5463–5474, 2020.

[25] Y. Ren, H. Guan, and Q. Zhao, "An efficient lattice-based linkable ring signature scheme with scalability to multiple layer," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 3, pp. 1547–1556, 2021.

[26] V. Lyubashevsky, "Fiat-Shamir with aborts: applications to lattice and factoring-based signatures," in *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*, pp. 598–616, Tokyo, Japan, December 2009.

[27] V. Lyubashevsky and D. Micciancio, "Generalized compact knapsacks are collision resistant," in *Proceedings of the 33rd International Conference on Automata Languages and Programming*, pp. 144–155, Venice, Italy, July 2006.

[28] C. A. Melchor, S. Bettaieb, X. Boyen, L. Fousse, and P. Gaborit, "Adapting Lyubashevsky's signature schemes to the ring signature setting," in *Proceedings of the 6th International Conference on Cryptology in Africa (AFRICACRYPT)*, pp. 1–25, Cairo, Egypt, June 2013.

[29] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[30] M. Abe, M. Ohkubo, and K. Suzuki, "1-out-of-n signatures from a variety of keys," *IEICE Transactions*, vol. 87, no. 1, pp. 131–140, 2004.

[31] K. Ohta and T. Okamoto, "On concrete security treatment of signatures derived from identification," in *Proceedings of the Annual International Cryptology Conference*, pp. 354–369, Santa Barbara CA USA, Augest1998.

[32] Z. Liu, K. Nguyen, G. Yang, H. Wang, and D. S. Wong, "A Lattice-Based Linkable Ring Signature Supporting Stealth Addresses," in *Proceedings of the 24th European Symposium on Research in Computer Security*, pp. 726–746, Luxembourg, September2019.

[33] V. Shoup, "NTL: A Library for Doing Number Theory," *Version 11.5.0*, https://www.shoup.net/ntl, 2021.