sensors

MDPI

Article

PSDAAP: Provably Secure Data Authenticated Aggregation Protocols Using Identity-Based Multi-Signature in Marine WSNs

Lifei Wei [1], Lei Zhang [1,*], Dongmei Huang [1], Kai Zhang [2], Liang Dai [1] and Guojian Wu [1]

[1] College of Information Technology, Shanghai Ocean University, Shanghai 201306, China; Lfwei@shou.edu.cn (L.W.); dmhuang@shou.edu.cn (D.H.); dailiang19931020@163.com (L.D.); wuguojian19930913@163.com (G.W.)

[2] Department of Computer Science and Technology, East China Normal University, Shanghai 200241, China; 52141201001@stu.ecnu.edu.cn

* Correspondence: Lzhang@shou.edu.cn; Tel.: +86-21-61900625

Received: 21 July 2017; Accepted: 6 September 2017; Published: 14 September 2017

Abstract: Data authenticated aggregation is always a significant issue for wireless sensor networks (WSNs). The marine sensors are deployed far away from the security monitoring. Secure data aggregation for marine WSNs has emerged and attracted the interest of researchers and engineers. A multi-signature enables the data aggregation through one signature to authenticate various signers on the acknowledgement of a message, which is quite fit for data authenticated aggregation marine WSNs. However, most of the previous multi-signature schemes rely on the technique of bilinear pairing involving heavy computational overhead or the management of certificates, which cannot be afforded by the marine wireless sensors. Combined with the concept of identity-based cryptography, a few pairing-free identity-based multi-signature (IBMS) schemes have been designed on the basis of the integer factorization problem. In this paper, we propose two efficient IBMS schemes that can be used to construct provably secure data authenticated aggregation protocols under the cubic residue assumption, which is equal to integer factorization. We also employ two different methods to calculate a cubic root for the cubic residue number during the signer's private key extraction. The algorithms are quite efficient compared to the previous work, especially for the algorithms of the multi-signature generation and its verification.

Keywords: identity-based multi-signature; provably secure; integer factorization; data authenticated aggregation; marine WSNs

1. Introduction

In most of the wireless sensor networks (WSNs), the significant issue for data collection or data aggregation always lies in the center of data transmission, both in the academia and in the industry [1–3]. In most scenarios of marine WSNs, all the nearby wireless sensors send their data, such as the temperature, pressure, salinity, and potential of hydrogen (pH value) in the chemistry of the environmental monitoring ocean, to a central node, which is located at a base station or a buoy for data collection, as shown in Figure 1. The central node further sends the aggregated data through the long-distance data transmission networks, such as vessel-based or satellite-based networks [4]. However, marine sensors are always deployed far away from the security monitoring. Thus, the secure data aggregation for marine sensor networks has emerged and attracted the interest of researchers and engineers. In order to mitigate the malicious attackers injecting false data, it is quite necessary for each central node to authenticate these sensing measurements from the nearby sensors in the ocean observation system [5].

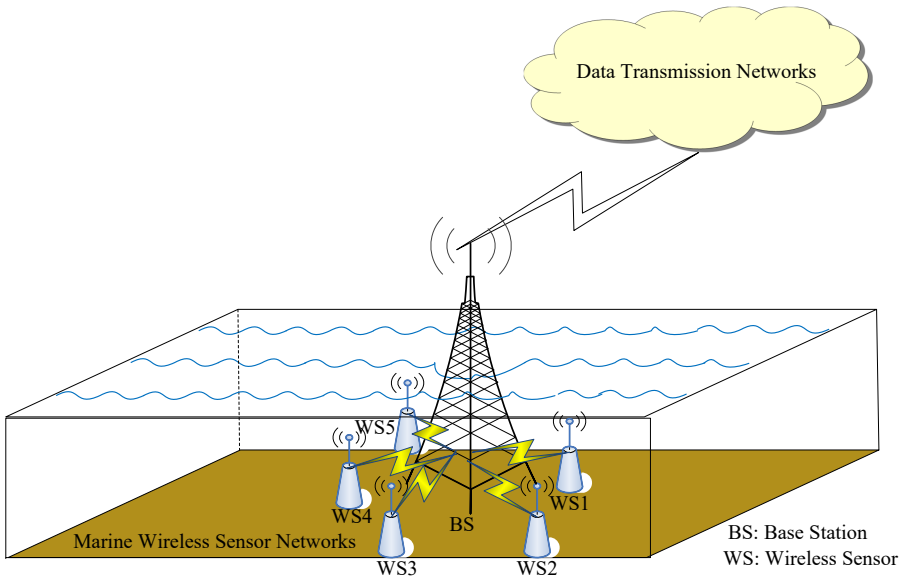

Figure 1. Data collection in marine wireless sensor networks (WSNs).

Generally, a digital signature often provides the properties of authenticity and non-repudiation through checking the signed acknowledgments from senders [6]. However, in WSNs, the international standards for broadcasting authentication are very vulnerable to signature verification flooding attacks, as the excessive requests for signature verification must run out of the computational resources of those victims [7]. The scenario seems worse, as the marine wireless sensors are powered by a limited battery and cannot afford these overloaded requests in an oceanic environment. To optimize the communication and computational overhead, a variant of digital signature, named *multi-signature*, permits various signers to sign on a message individually and aggregate partial signatures to a compact signature [8].

A multi-signature can play a significant role in authenticating different sensors' data by checking a single compact signature to cut down the communication bandwidth for marine wireless devices, as the transmission of one-bit data consumes more energy than the arithmetic operations on several bits [9]. This seems a promising way to solve the data authentication in a multi-user scenario. Since the primitive has been proposed, multi-signature schemes have been paid attention to by most of the network designers and industry engineers. However, in the past years, most of the work on multi-signature schemes has been constructed by relying on the assumed existence of *public key infrastructure* (**PKI**) [10,11]; the heavy burdens of the digital public key certificate management bring high communication overhead and storage overhead when **PKI** is applied and implemented in the wireless networks. The cases become worse when the sensors are deployed in the marine environments (denoted as **Problem 1**).

To overcome the weakness brought by **PKI**, identity-based cryptography emerges as a novel cryptographic primitive and a powerful alternative to traditional certificate-based cryptography, which has been raised early on in [12] and is further specifically designed in [13,14]. Identity-based cryptography makes some public, known information a public key, such as the device's number, IP address, or a username, to mitigate the management problem for the public key certificates. In the extreme case that the bandwidth is a bottleneck, the identities of the signers often appear in the head of the communication packets, instead of in the transmission of the heavy public keys. Inspired by this concept, the first *identity-based multi-signature* (**IBMS**) scheme, proposed in [15], uses a mathematical technique named "bilinear mapping", such as is used in [13], and is proved to be secure, relying on *discrete logarithm* (DL) assumptions or *computational Diffie–Hellman* (CDH) assumptions. Because the operation of bilinear mapping involves too much computational overhead [16,17], many bilinear

mapping techniques are not suitable for the battery-limited sensors in marine WSNs (denoted as **Problem 2**).

As a consequence, there is great interest for cryptographic researchers to design pairing-free identity-based cryptographic schemes [18]. The first non-pairing IBMS scheme was proposed in [19] with three-round interactive communications and under R. Rivest, A. Shamir, L. Adleman (RSA) assumptions. Later, a communication efficiency-improved IBMS scheme under RSA assumptions was presented in [20] with two-round interactive communications. Yang et al. [21] proposed an efficient improved IBMS scheme that aims to save the computational resources and communication bandwidth. Even if the RSA assumption approaches the integer factorization assumptions, unfortunately, the RSA assumption has not yet been proved equal to the factorization assumption (denoted as **Problem 3**).

To satisfy the application requirements and to avoid security concerns in cryptrography, it is common practice to construct alternative cryptographic schemes under a weaker assumption—integer factorization. Recently, cryptographic researchers have been focused on finding a new construction that is proved to be secure directly on the basis of factorization. Chai [22] gave an instance of an identity-based digital signature relying on the quadratic residue assumption. Following this, Wei et al. [6] proposed IBMS schemes using quadratic residue assumptions, under weaker assumptions and a strengthened security model, achieving advantages in the computational consumption and transmission overhead. Xing [23] and Wang [24] presented identity-based signature schemes under the cubic residue assumptions. Wang proposed several signature variants relying on cubic residues, including identity-based ring signature [25], *identity-based proxy multi-signature* (**IBPMS**) [26] and threshold ring signature [27]. Wei [28] considered an *identity-based multi-proxy signature* (**IBMPS**) scheme for use in a cloud-based data authentication protocol. Zhang [29] proposed a secure multi-entity delegated authentication protocol based on an *identity-based multi-proxy multi-signature* (**IBMPMS**) for mobile cloud computing. Unfortunately, none considered constructing IBMS schemes directly based on cubic residues (denoted as **Problem 4**).

Facing the above problems, this work constructs IBMS schemes relying on the cubic residue assumption equal to integer factoring. Our schemes have merits not only in the efficiency aspect, where we do not rely on the bilinear pairing maps or over exponentiations, but also in the security aspect, where we prove them to be secure under a weaker assumption of factoring to achieve stronger security. The contributions for this paper can be summarized as follows.

1. We have proposed two efficient IBMS schemes, denoted as $\mathbf{IBMS^{CR}-1}$ and $\mathbf{IBMS^{CR}-2}$, which are suitable for data aggregation among the sensors and collectors in marine WSNs.
2. We formally define the security of IBMS and prove $\mathbf{IBMS^{CR}-1}$ to be secure, relying on the cubic residues in a random oracle model. The computational cost of $\mathbf{IBMS^{CR}-1}$ is lower, as the exponentiations are cubic exponentials.
3. To enhance efficiency, the total computational cost of $\mathbf{IBMS^{CR}-2}$ is almost four-fifths that of $\mathbf{IBMS^{CR}-1}$ in implementation. We also prove the security of $\mathbf{IBMS^{CR}-2}$ on the basis of the cubic residues equalling integer factoring in the random oracle model.

The organization of this paper is as follows. Section 2 gives necessary preliminaries, and Section 3 gives the formal definition of the security model. In Sections 4 and 5, we propose two concrete IBMS schemes, $\mathbf{IBMS^{CR}-1}$ and $\mathbf{IBMS^{CR}-2}$, as well as outline their correctness and full security proof. Section 6 gives the performance comparison. Section 7 gives the conclusion for the paper.

2. Preliminaries

Some fundamental concepts are introduced simply, for further explaining the construction and security proof.

2.1. Cubic Residue

We first introduce the definition of the cubic residue.

Definition 1 (Cubic residue [23]). *For an integer $N \equiv 1 \pmod 3$, a cubic residue modulo N, $c \in \mathbb{Z}_N^*$, if $x^3 \equiv c \pmod N$ for some $x \in \mathbb{Z}_N^*$.*

Because the module N is a product for unknown p and q, it is difficult to obtain x from a cubic residue c, that is, the difficulty of obtaining x from c is equal to the factorization of N.

2.2. Cubic Residue Symbol in Eisenstein Ring

Following the work in [23,30,31], we let ω denote a complex root of $z^2 + z + 1 = 0$, which means that ω is a cubic root of 1. We also have $\omega^2 = -1 - \omega = \bar{\omega}$, where $\bar{\omega}$ is the conjugate complex of ω. The Eisenstein ring is defined as the set $\mathbb{Z}[\omega] = \{a + b\omega | a, b \in \mathbb{Z}\}$. We introduce the cubic residue symbol as follows:

$$\left(\frac{\cdot}{\cdot}\right)_3 : \mathbb{Z}[\omega] \times (\mathbb{Z}[\omega] - (1-\omega)\mathbb{Z}[\omega]) \to \{0, 1, \omega, \omega^2\}$$

For a prime p in $\mathbb{Z}[\omega]$ where p is not associated to $1 - \omega$, we have

$$\left(\frac{\alpha}{p}\right)_3 = \alpha^{(N(p)-1)/3} \pmod p$$

where $N(p) = p \cdot \bar{p}$ is defined as the norm of p.

2.3. Some Useful Theorems

Theorem 1 (Factorization Theorem [23]). *Let $N = pq$, where p and q are large primes. Let c be a cubic residue modulo N, and r_1 and r_2 be c's two cubic roots modulo N; that is, $r_1^3 \equiv r_2^3 \equiv c \pmod N$ and $r_1 \neq r_2 \pmod N$. N can be factored by taking $\gcd(r_1 - r_2, N)$ in polynomial time, where $\gcd(x, y)$ is the greatest common divisor of x and y.*

Theorem 1 is easily validated, as if $r_1^3 \equiv r_2^3 \equiv c \pmod N$, we have $(r_1 - r_2)(r_1^2 + r_1 r_2 + r_2^2) \equiv 0 \pmod N$. There must exist an integer k such that $(r_1 - r_2)(r_1^2 + r_1 r_2 + r_2^2) = kpq$. If $r_1 \neq r_2 \pmod N$, $r_1 - r_2$ cannot be a multiple of N at the same time; $r_1 - r_2$ must contain a non-trivial divisor of N, which is p or q. Therefore, the integer N can be factored by Theorem 1. However, the two cubic roots satisfying $r_1 \equiv r_2 \pmod N$ cannot lead directly to factoring the integer N.

The following theorem shows a solution to compute a 3^ℓ-th root of a cubic residue without factoring N.

Theorem 2. *Let $\omega \equiv 1 \pmod 3$, $\ell > 0$, c be a cubic residue modulo N, and $X \in Z_N^*$ satisfy*

$$c^\omega \equiv X^{3^\ell} \pmod N$$

Then we can easily calculate the cubic root y; that is, $y^3 \equiv c \pmod N$.

Because $\omega \equiv 1 \pmod 3$, we can denote $\omega = 3^r(3\delta + 1)$; following this,

$$c^\omega \equiv c^{3^r(3\delta+1)} \equiv X^{3^\ell} \pmod N$$

We take the 3^r-th root and obtain

$$c^{3\delta+1} \equiv X^{3^\ell - r} \pmod N$$

Because $c^{3\delta+1} = c^{3\delta} \cdot c$, we have

$$c \equiv \frac{X^{3^\ell - r}}{c^{3\delta}} \equiv \left(\frac{X^{3^\ell - r - 1}}{c^\delta}\right)^3 \pmod N.$$

Let $y = X^{3^{\ell}-r-1}/c^{\delta}$; then we have $y^3 \equiv c \pmod{N}$

Theorem 2 can be used in the security proof for **IBMS$^{\mathbf{CR}}$−1**. We introduce the following Theorem [24,29] regarding the cubic residue used in the security proof for **IBMS$^{\mathbf{CR}}$−2**.

Theorem 3 (Cubic residue construction [24,29])**.** *If p and q are two primes with $p \equiv 2 \pmod 3$ and $q \equiv 4$ or $7 \pmod 9$, it is easy to produce a cubic residue modulo N. Let nc be a non-cubic modulo q, for any $h \in \mathbb{Z}_N^*$; we can compute that $\eta = \frac{(q-1) \pmod 9}{3}$, $\lambda = \eta \pmod 2 + 1$, $\beta = (q-1)/3$, $\xi \equiv nc^{\eta\beta} \pmod q$, $\tau \equiv h^{\lambda\beta} \pmod q$ and*

$$b = \begin{cases} 0, & \text{if } \tau = 1 \\ 1, & \text{if } \tau = \xi \\ 2, & \text{if } \tau = \xi^2 \end{cases}$$

We can construct a cubic residue C modulo N; that is, $C = nc^b \cdot h \pmod N$.

Theorem 4. *Let p, q, N, C, and η be defined as in Theorem 3; we can calculate a cubic root s of C^{-1} by $s \equiv C^{[2^{\eta-1}(p-1)(q-1)-3]/9} \pmod N$. Note that $s^3 \cdot C \equiv 1 \pmod N$.*

3. Formal Definition and Security Model

3.1. Formal Definition

We assume that there exist n distinct signers, named $ID_1, ID_2, ..., ID_n$, to authenticate a message m by cooperatively generating a multi-signature $m\sigma$. The signer ID_i is denoted as $signer_i$.

Theorem 5. *A typical IBMS scheme is always made up of six algorithms, that is, **Setup**, **Extra**, **Sign**, **Verify**, **MSign**, and **MVerify**. We describe each of them as follows.*

- ***Setup**: $(mpk, msk) \leftarrow$ **Setup** (1^k). The algorithm is controlled by the* key generator center *(**KGC**). The **KGC** generates the system's master public keys mpk and master secret keys msk when it is given the security parameter k.*
- ***Extra**: $sk_{ID} \leftarrow$ **Extra** (mpk, msk, ID). The algorithm is also controlled by the **KGC**, given msk, mpk and a user's identity ID, such as a string. It returns the private key sk_{ID} through secure channels.*
- ***Sign**: $\sigma \leftarrow$ **Sign** (mpk, sk, m, ID): The signer uses its private key sk, the identity ID, and the message to be signed m to generate a signature σ on m.*
- ***Verify**: $\{0,1\} \leftarrow$ **Verify** (mpk, ID, m, σ): The algorithm takes the signer's identity ID, the data m, and a candidate signature σ. If σ is a valid signature, it returns 1. Otherwise, it returns 0.*
- ***MSign**: $m\sigma \leftarrow$ **MSign** $(mpk, sk, m, IDSet)$. The signer with the private sk joins in the multi-signing algorithm, which needs additional parameters, including a message m and an identity set $IDSet = \{ID_1, ID_2, ..., ID_n\}$ containing all the identities of the signers. After several rounds of interactive communication, **MSign** generates a multi-signature $m\sigma$.*
- ***MVerify**: $\{0,1\} \leftarrow$ **MVerify** $(mpk, IDSet, m, m\sigma)$. The algorithm returns 1 if $m\sigma$ is a valid multi-signature on the message m by authenticating the signers in $IDSet$.*

Correctness. When all of the participating signers honestly and correctly execute the algorithm **MSign** using the private keys, derived from the algorithm **Extra**, each of the signers will end the algorithm by obtaining a local multi-signature $m\sigma$ such that

$$\mathbf{MVerify}(IDSet, m, m\sigma, mpk) = 1$$

where all mpk and msk are generated by the algorithm **Setup** and $IDSet$ includes n identities $ID_1, ID_2, ..., ID_n$ for any messages $m \in \{0,1\}^*$.

3.2. Security Model

This considers an extreme case: the adversary $\mathcal{A}$ compromising the $n-1$ participants and leaving *only one* honest user, denoted $signer_1$. The $signer_1$ user is controlled by the challenger $\mathcal{C}$. When the game starts, $\mathcal{C}$ gives $\mathcal{A}$ the honest identity of $signer_1$ and allows $\mathcal{A}$ to compromise the other signers' private keys. It also assume that a secure channel between the signers is not guaranteed. All of the communication among the signers can be eavesdropped upon. $\mathcal{C}$ provides $\mathcal{A}$ a hash oracle, a key extraction oracle and a multi-sign oracle. $\mathcal{A}$'s final target is to successfully forge a multi-signature.

Definition 2. *Considering the games between $\mathcal{A}$ and $\mathcal{C}$.*

- ***Setup**: $\mathcal{C}$ executes the algorithm to generate the master public keys mpk and sends mpk to $\mathcal{A}$.*
- ***Query**: $\mathcal{A}$ is allowed to query to $\mathcal{C}$ in an adaptive way.*
  - ***Extraction-query** (mpk, ID). $\mathcal{C}$ executes **Extra** to obtain sk_{ID} and sends to $\mathcal{A}$ when $\mathcal{A}$ asks for the private key of $signer_{ID}$.*
  - ***Multi-signature query** $(mpk, m, IDSet)$. $\mathcal{C}$ obtains a multi-signature $m\sigma$ and sends to $\mathcal{A}$ when $\mathcal{A}$ asks for the multi-signature $m\sigma$ on m and $IDSet$.*
  - ***Hash-query**. $\mathcal{C}$ chooses the returned values by itself and sends to $\mathcal{A}$ when $\mathcal{A}$ asks.*
- ***Forgery**. $\mathcal{A}$ makes a multi-signature as a forgery, that is, $m\sigma^*$ on m^* for $IDSet^*$, which contains at least one uncompromised user's identity; meanwhile, $\mathcal{A}$ never sends $(mpk, IDSet^*, m^*)$ to the multi-signature query.*

Definition 3 (Attack Goals). *The advantage $Adv_{\mathcal{A}}^{\boldsymbol{IBMS}}$ in breaking the $\boldsymbol{KG}(k)$ problems is defined as*

$$Adv_{\mathcal{A}}^{\boldsymbol{IBMS}}(k) = \Pr\left[x^{3^{\ell}} \equiv y \pmod{N} \;\middle|\; \begin{array}{c} (N,p,q) \leftarrow \boldsymbol{KG}(k) \\ y \leftarrow \mathbb{Z}_N^* \\ x \leftarrow \mathcal{A}(N,\ell,y) \end{array} \right]$$

Definition 4 (Unforgeability). *An adversary $\mathcal{A}$ $(t, q_H, q_E, q_S, n, \epsilon)$ breaks the scheme if $\mathcal{A}$ executes for a time of t at most, and makes at most q_H hash queries, q_E extraction queries, and q_S multi-signature queries with n participants, and $Adv_{\mathcal{A}}$ is at least ϵ. An IBMS scheme $(t, q_E, q_S, q_H, n, \epsilon)$ has unforgeability if there exists no attacker $\mathcal{A}$ $(t, q_H, q_E, q_S, n, \epsilon)$ that breaks it.*

4. Concrete Construction of IBMS$^{\mathbf{CR}}$-1

4.1. Construction

Inspired by the previous work [6,22,23], we propose a concrete identity-based multi-signature scheme (**IBMS$^{\mathbf{CR}}-$1**) with three-round interactive communications among the marine sensors and the generation of a single multi-signature as an authenticated tag.

- `Setup` (k, ℓ): The key generator center inputs security parameters k and ℓ, and then:
  1. Chooses two random primes p and q, such that $p \equiv q \equiv 1 \pmod 3$ and $(p-1)(q-1)/9 \equiv -1 \pmod 3$. Without loss of generality, we assume that $(p-1)/3 \equiv -1 \pmod 3$, $(q-1)/3 \equiv 1 \pmod 3$.
  2. Chooses two random primes π_1 and π_2 from the Eisenstein ring $\mathbb{Z}[\omega]$, *s.t.* the norms satisfy $N(\pi_1) = p$ and $N(\pi_2) = q$.
  3. Computes $N = p * q$. We let $A + B\omega = \pi_1\pi_2$, $A, B \in \mathbb{Z}$, and then compute $C = -AB^{-1} \pmod N$. Note that $\left(\frac{C}{p}\right)_3 = \omega^2$, and $\left(\frac{C}{q}\right)_3 = \omega$.
  4. Chooses a random number $a \in \mathbb{Z}_N^*$ such that $\left(\frac{a}{N}\right)_3 = \omega$.
  5. Computes $d = \frac{1}{3}[\frac{1}{9}(p-1)(q-1)+1]$.

6. Selects three hash functions $h_1(\cdot)$, $h_2(\cdot)$, and $h_3(\cdot)$ such that $h_1(\cdot) : \{0,1\}^* \to \mathbb{Z}_N^*$, h_2 and $h_3(\cdot) : \{0,1\}^* \to \{0,1\}^\ell$.

As a result of the step **Setup**, the master secret key is $msk = (p, q, d)$, which is securely stored, and the public parameter is $mpk = (N, h_1, h_2, h_3, a, C, \ell)$.

- `Extra` (mpk, msk, ID): **KGC** inputs the identity ID, computes the hash value of ID as $h_1(ID)$ and obtains a first symbol $c_{ID,1}$ such that

$$c_{ID,1} = \begin{cases} 0, & \text{if } \left(\frac{h_1(ID)}{N}\right)_3 = 1 \\ 1, & \text{if } \left(\frac{h_1(ID)}{N}\right)_3 = \omega^2 \\ 2, & \text{if } \left(\frac{h_1(ID)}{N}\right)_3 = \omega \end{cases}$$

We let $h = a^{c_{ID,1}} \cdot h_1(ID)$ and we have $\left(\frac{h}{N}\right)_3 = 1$. Following this, **KGC** computes a second symbol $c_{ID,2}$ such that

$$c_{ID,2} = \begin{cases} 0, & \text{if } \left(\frac{h}{p}\right)_3 = \left(\frac{h}{q}\right)_3 = 1 \\ 1, & \text{if } \left(\frac{h}{p}\right)_3 = \omega, \left(\frac{h}{q}\right)_3 = \omega^2 \\ 2, & \text{if } \left(\frac{h}{p}\right)_3 = \omega^2, \left(\frac{h}{q}\right)_3 = \omega \end{cases}$$

We let $I_{ID} = C^{c_{ID,2}} \cdot a^{c_{ID,1}} \cdot h_1(ID)$. It is easy to find that $I_{ID} \in \mathbb{CR}_N$, as $\left(\frac{I_{ID}}{p}\right)_3 = \left(\frac{I_{ID}}{q}\right)_3 = 1$. Finally, **KGC** extracts the private key sk_{ID} as a 3^ℓ-th root of I_{ID}:

$$sk_{ID} \equiv I_{ID}^{d^\ell} \pmod{N} \tag{1}$$

KGC sends sk_{ID} as well as $(c_{ID,1}, c_{ID,2})$ to signer $_{ID}$ secretly. Note that $I_{ID} \equiv sk_{ID}^{3^\ell} \pmod{N}$. Following this, we denote $\widetilde{ID} = \{ID, c_{ID,1}, c_{ID,2}\}$.

- `Sign` and `verify`: These two algorithms can be derived from [23].
- `MSign` $(mpk, sk_1, m, IDSet)$: For simplicity, **IBMS$^{\mathbf{CR}}$−1** is described from the MS_1's point of view. Given the MS_1's private key sk_1, the message m and the identity set $IDSet = \{\widetilde{ID_1}, \widetilde{ID_2}, ..., \widetilde{ID_n}\}$, MS_1 executes the following algorithm from Algorithm 1. **MSign** generates $m\sigma = (w, u)$ as the multi-signature.
- `MVerify` $(mpk, IDSet, m, m\sigma)$. The algorithm verifies by the following three steps.

(1) For $i = 1, 2, ..., n$, it computes $I_i \equiv C^{c_{ID_i,2}} \cdot a^{c_{ID_i,1}} \cdot h_1(ID_i) \pmod{N}$.
(2) It computes $\hat{R} \equiv u^{3^\ell} \left(\prod_{i=1}^n I_i\right)^{-w} \pmod{N}$.
(3) It checks whether

$$w = h_3(\hat{R} \| IDSet \| m) \tag{2}$$

is satisfied. If Equation (2) is satisfied, **MVerify** returns 1. This means $m\sigma$ is valid. Otherwise **MVerify** returns 0.

4.2. Correctness

The correctness follows:

$$u^{3^\ell} \equiv \prod_{i=1}^n u_i^{3^\ell} \equiv \prod_{i=1}^n r_i^{3^\ell} sk_i^{w3^\ell} \equiv \prod_{i=1}^n R_i I_i^{(3d)^\ell w} \equiv R \prod_{i=1}^n I_i^w \pmod{N}$$

We have $\hat{R} \equiv R \equiv u^{3^\ell} \prod_{i=1}^n I_i^{-w} \pmod{N}$.

Algorithm 1: The MSign Algorithm in IBMS $^{CR}-1$.

Input: the master public key mpk, the private key sk, the identity set $IDSet$, the message to be signed m;
Output: a multi-signature $m\sigma$.

1. Each MS_i randomly selects $r_i \in \mathbb{Z}_N^*$ and computes $R_i \equiv r_i^{3^\ell} \pmod N$ and $t_i = h_2(R_i)$.
2. MS_i only broadcasts t_i to other signers MS_j ($j \neq i$) in $IDSet$ and keeps R_i temporarily.
3. After receiving t_i from MS_i ($2 \leq i \leq n$), MS_1 then broadcasts R_1 to other MS_i.
4. After receiving R_i from MR_i, MS_1 checks whether $t_i = h_2(R_i)$ for $2 \leq i \leq n$ is satisfied.
5. If one of these fails, the algorithm stops, which means the attackers have mixed invalid partial signatures. Otherwise, MS_1 sets $R \equiv \prod_{i=1}^{n} R_i \pmod N$, $w = h_3(R \| IDSet \| m)$, and $u_1 \equiv r_1 \cdot sk_1^w \pmod N$.
6. MS_1 broadcasts u_1 to other MS_i.
7. After receiving u_i from MS_i, MS_1 aggregates these by $u \equiv \prod_{i=1}^{n} u_i \pmod N$.
8. Each MS_i locally generates a multi-signature $m\sigma = (w, u)$.

Return $m\sigma$;

4.3. Security Proof

IBMS$^{CR}-1$ is provably secure under the factorization in the random oracle model.

Theorem 6. *If the factorization problem is (t', ϵ')-hard, **IBMS$^{CR}-1$** is $(t, q_E, q_H, q_S, n, \epsilon)$-secure against existential forgery attackers under the adaptively chosen message attack and chosen identity attack. We have estimates for t' and ϵ' as follows:*

$$\epsilon' \geqslant \frac{2\epsilon^2}{3(q_H+1)} - \left(\frac{2nq_Sq_H + n^2q_S^2 + q_H^2}{2^{\ell_R} \cdot (q_H+1)} + \frac{nq_S}{2^{\ell_0-1}} \right) \epsilon - \frac{1}{3 \cdot 2^{\ell-1}} \tag{3}$$

Proof. We assume $\mathcal{C}$ is given a factorization instance N for a product of unknown p and q, and obtain the result of p or q with a non-negligible probability. $\mathcal{C}$ plays with $\mathcal{A}$ as follows.

Firstly, $\mathcal{C}$ selects $a \in \mathbb{Z}_N^*$, such as a non-cubic residue and a secure parameter $\ell \geqslant 160$ (the length of ℓ has been discussed and suggested in [22]), and sends (N, a, ℓ) to $\mathcal{A}$ as mpk. $\mathcal{C}$ manages several lists: one signature list and three hash lists.

Then, $\mathcal{C}$ starts to answer according to $\mathcal{A}$'s queries, as follows.

- h_1**-Query** (ID): $\mathcal{C}$ manages a list $(ID, h_1, s, c_{ID,1}, c_{ID,2})$. When $\mathcal{A}$ requests the identity ID, $\mathcal{C}$ answers as h_1. $(c_{ID,1}, c_{ID,2}) \in \{0,1\}^2$ in two bits and $s \in \mathbb{Z}_N^*$ is used as a secret key. When $\mathcal{A}$ asks on ID, $\mathcal{C}$ answers h_1 if ID has existed in the h_1-list. Otherwise, $\mathcal{C}$ randomly selects $s \in \mathbb{Z}_N^*$ and $(c_{ID,1}, c_{ID,2}) \in \{0,1\}^2$, calculates

$$h_1 \equiv \frac{s^{3^\ell}}{(-1)^{c_{ID,2}} \cdot (a)^{c_{ID,1}}} \pmod N \tag{4}$$

  and returns the answer h_1 to $\mathcal{A}$, adding $(ID, h_1, s, c_{ID,1}, c_{ID,2})$ to the h_1-list.
- h_2**-Query** (R): $\mathcal{C}$ manages a list (R, h_2). When $\mathcal{A}$ asks on R, $\mathcal{C}$ answers h_2 if R has existed in the h_2-list. Otherwise, $\mathcal{C}$ randomly selects $h_2 \in \{0,1\}^{\ell_0}$, adds (R, h_2) into the h_2-list and returns h_2.
- h_3**-Query** $(R, m, IDSet)$: $\mathcal{C}$ manages a list $(R, m, IDSet, h_3)$. When $\mathcal{A}$ asks on $(R, m, IDSet)$, $\mathcal{C}$ returns h_3 if $(R, m, IDSet)$ has existed in the h_3-list. Otherwise, $\mathcal{C}$ randomly selects $h_3 \in \mathbb{Z}_N^*$, returns h_3, and adds $(R, m, IDSet, h_3)$ to the h_3-list.
- **Extraction query** (ID): $\mathcal{C}$ executes an additional h_1-query if ID does not yet exist in the h_1-list and returns s and $(c_{ID,1}, c_{ID,2})$.
- **Multi-signature queries**: $\mathcal{C}$ checks in the h_1-list for whether ID_1 exists. If ID_1 is already in the h_1-list, $\mathcal{C}$ has obtained the private key of $signer_1$ and simulates the game as the real algorithm

MSign $(sk_1, IDSet, m)$ using the secret key $sk_1 = s_1$. Otherwise, $\mathcal{C}$ does not have the private key of $signer_1$ and executes the following steps:

- $\mathcal{C}$ plays as $signer_1$, and randomly chooses $t_1 \leftarrow \{0,1\}^{\ell_0}$, broadcasting t_1 to other signers. $\mathcal{C}$ also waits to receive $t_2, t_3, \cdots, t_n$ from others; it randomly selects $w \leftarrow \{0,1\}^{\ell}$ and $u_1 \leftarrow \mathbb{Z}_N^*$, and calculates
$$R_1 = u_1^{3^\ell} \left((-1)^{c_{ID_1,2}} \cdot a^{c_{ID_1,1}} \cdot h_1(ID_1)\right)^{-w} \tag{5}$$
If R_1 already exists in the h_2-list, $\mathcal{C}$ stops. Otherwise, $\mathcal{C}$ sets (R_1, t_1) in the h_2-list. $\mathcal{C}$ looks up R_i such that (R_i, t_i) where $2 \leqslant i \leqslant n$. If for some i the record is found, $\mathcal{C}$ also stops. Otherwise, $\mathcal{C}$ calculates $R = \prod_{i=1}^{n} R_i \pmod N$ and sets $h_3(R\|S\|m) = w$, or stops if the entry has already existed.
- $\mathcal{C}$ sends R_1 to other signers. After receiving $R_2', \cdots, R_n'$ from the signers, $\mathcal{C}$ verifies that $h_2(R_i') \stackrel{?}{=} t_i$. $\mathcal{C}$ ends up with the protocol if one of these does not satisfy this, which means $\mathcal{A}$ has to guess the results of the hash value. If $R_i \neq R_i'$ for some i, $\mathcal{C}$ stops. $\mathcal{C}$ sends u_i to the signers, receives $u_2, u_3, \cdots, u_n$, and calculates $u = \prod_{i=1}^{n} u_i \pmod N$. Finally, $\mathcal{C}$ sends $m\sigma = (w, u)$ to $\mathcal{A}$.

At the end of the game, $\mathcal{A}$ generates a multi-signature $m\sigma^* = (w^*, u^*)$ on message m^*. $\mathcal{C}$ calculates

$$R^* \leftarrow (u^*)^{3^\ell} \prod_{i=1}^{n} \left((-1)^{c_{ID_i^*,2}} a^{c_{ID_i^*,1}} h_1(ID_i^*)\right)^{-w^*} \tag{6}$$

and makes an additional query $h_3(R^*\|IDSet^*\|m^*)$. We let $U \subseteq IDSet^* = \{ID_1^*, ID_2^*, ..., ID_n^*\}$ denote the honest IDSet, that is, $\mathcal{A}$ never compromised. If $\mathcal{A}$ succeeded in forgery, that is,

- **MVerify** $(mpk, IDSet^*, m^*, \sigma^*) = 1$
- $U \neq \emptyset$
- $\mathcal{A}$ has never queried $(IDSet^*, m^*)$ to the signature oracle

then $\mathcal{C}$ checks the h_1-list. If the multi-signature is valid, we can obtain

$$\begin{aligned} u^{*3^\ell} &\equiv R^* \prod_{i=1}^{n} \left((-1)^{c_{ID_i,2}} a^{c_{ID_i,1}} h_1(ID_i^*)\right)^{w^*} \\ &\equiv R^* \prod_{i=1}^{n} s_i^{*3^\ell w^*} \pmod N \end{aligned} \tag{7}$$

We let $s^* \leftarrow \prod_{i=1}^{n} (s_i^*)^{3^\ell} \pmod N$ and produce (s^*, σ^*).

To factor N by applying the rewinding technique, $\mathcal{C}$ plays with $\mathcal{A}$ once again using the random tapes, which are the same as for the first time. Because $\mathcal{C}$ previously recorded the transcripts, $\mathcal{C}$ obtains the same results for $\mathcal{A}$'s queries.

When $\mathcal{A}$ queries for h_3, $\mathcal{C}$ randomly selects an alternative answer w' instead of w, as, in the second run, the h_1- and h_2-query are equal to those of the first round.

$\mathcal{C}$ generates $(s, m\sigma)$ and $(s', m\sigma')$ such that

$$u^{3^\ell} \equiv Rs^w \ \ and \ \ u'^{3^\ell} \equiv R's'^{w'}$$

By $R = R'$, $m = m'$ and $s = s'$, we have

$$\left(\frac{u}{u'}\right)^{3^\ell} \equiv s^{(w-w')} \pmod N \tag{8}$$

Because $w \neq w' \in \{0,1\}^{\ell_0}$ and $\ell_0 < \ell$, we can obtain $|w - w'| < 3^{\ell}$. According to Theorem 2, $\mathcal{C}$ can calculate a cubic root $\tilde{s}$ where $\tilde{s}^3 = s$. Meanwhile, $\mathcal{C}$ checks the h_1-list to search for an entry in which $ID_i \in IDSet$ and calculates $\bar{s} = \prod_{i \in IDSet} s_i^{3^{\ell-1}}$.

Therefore, $\tilde{s}^3 \equiv \bar{s}^3 \equiv s \pmod{N}$. If $\bar{s} \neq \tilde{s} \pmod{N}$, N can be factored by Theorem 1. Otherwise, $\mathcal{C}$ cannot factor N. The probability that $\tilde{s} \neq \bar{s} \pmod{N}$ is 2/3.

Finally, we calculate the probability that $\mathcal{C}$ returns a *valid* result. Because most of the simulation game is similar to in [6], we set ϵ', ϵ and ϵ^* as the probability to factor N by $\mathcal{C}$, the probability to forge a multi-signature in practice by $\mathcal{A}$ and the probability to succeed in the first run before the rewinding technique by $\mathcal{A}$, respectively.

We have

$$\epsilon^* \geqslant \epsilon - \frac{q_S(q_H + nq_S)}{2^{\ell_N}} - \frac{(q_H + nq_S)^2}{2^{\ell_N+1}} - \frac{2q_S(q_H + q_S)}{2^{\ell_N}} - \frac{nq_S}{2^{\ell_0}} \tag{9}$$

Furthermore, according to the forking lemma [32], we can easily obtain

$$frk \geqslant \epsilon^* \left(\frac{\epsilon^*}{q_H} - \frac{1}{2^{\ell}} \right) \geqslant \frac{\epsilon^{*2}}{q_H + 1} - \frac{1}{2^{\ell}} \tag{10}$$

The probability that $\mathcal{C}$ succeeds to factor N is

$$\begin{aligned} \epsilon' &\geqslant \frac{2}{3} \cdot frk \geqslant \frac{2\epsilon^{*2}}{3(q_H + 1)} - \frac{1}{3 \cdot 2^{\ell-1}} \\ &\geqslant \frac{2\epsilon^2}{3(q_H + 1)} - \left(\frac{2nq_Sq_H + n^2q_S^2 + q_H^2}{2^{\ell_R} \cdot (q_H + 1)} + \frac{nq_S}{2^{\ell_0-1}} \right) \epsilon - \frac{1}{3 \cdot 2^{\ell-1}} \end{aligned} \tag{11}$$

□

5. Concrete Construction of IBMS$^{CR}-2$

Inspired by the related work [24,26,29], we give a more efficient IBMS construction (named **IBMS$^{CR}-2$**), whose computational overhead in **MSign** and **MVerify** is much lower than for those in **IBMS$^{CR}-1$**.

5.1. Construction

- **Setup** (k, ℓ): Given the security parameters, **Setup** can be executed as follows.

  (1) **KGC** chooses random primes p and q where $p \equiv 2 \pmod{3}$ and $q \equiv 4$ or $7 \pmod{9}$, and calculates the product $N = p \cdot q$.

  (2) A non-cubic residue a is selected such that $\left(\frac{a}{q}\right) = -1$.

  (3) Several computational parameters are computed:

$$\begin{aligned} \eta &= [q - 1 \pmod{9}]/3 \\ \lambda &= \eta \pmod{2} + 1 \\ \beta &= (q - 1)/3 \\ \xi &= a^{\eta\beta} \pmod{q} \end{aligned}$$

  (4) Three hash functions h_1, h_2 and h_3 are picked up, where h_1:$\{0,1\}^* \rightarrow Z_N^*$, h_2, h_3:$\{0,1\}^* \rightarrow \{0,1\}^{\ell}$.

  Finally, the algorithm **Setup** outputs $msk = (p, q, \beta)$ and $mpk = (N, h_1, h_2, h_3, a, \eta, \lambda)$. **KGC** keeps msk secretly.

- **Extra** (mpk, msk, ID): **KGC** computes sk as follows:

(1) **KGC** computes $\omega = h_1(ID)^{\lambda\beta} \pmod q$ and set sa symbol c_{ID} according to ω and ξ:

$$c_{ID} = \begin{cases} 0, & \text{if } \omega = 1 \\ 1, & \text{if } \omega = \xi \\ 2, & \text{if } \omega = \xi^2 \end{cases}$$

KGC denotes $I = a^{c_{ID}} \cdot h_1(ID) \pmod N$.

(2) **KGC** calculates

$$sk = I^{\frac{2^\eta (p-1)(q-1)-3}{9}} \pmod N \tag{12}$$

and securely distributes sk to the signer. We have $sk_i^3 \cdot I_i \equiv 1 \pmod N$. Following this, we denote the identity by $\widetilde{ID_i} = \{ID_i, c_{ID_i}\}$.

- **Sign** and **verify**: These two algorithms can be derived from [29].
- **MSign** $(mpk, sk_1, m, IDSet)$: Given the MS_1's private key sk_1, the message m and the identity set $IDSet = \{\widetilde{ID_1}, \widetilde{ID_2}, ..., \widetilde{ID_n}\}$, MS_1 executes the following algorithm in Algorithm 2. **MSign** generates the multi-signature $m\sigma = (w, u)$.
- **MVerify** $(mpk, IDSet, m, m\sigma)$. The algorithm verifies by the following three steps:

(1) For $i = 1, 2, ..., n$, it computes $I_i = a^{c_{ID_i}} \cdot h_1(ID_i)$.

(2) It computes $\hat{R} = u^3 \cdot (\prod_{i=1}^n I_i)^w \ (mod\ N)$.

(3) It checks whether

$$w = h_3(\hat{R} \| IDSet \| m) \tag{13}$$

is satisfied. If Equation (13) is satisfied, **MVerify** returns 1. This means $m\sigma$ is valid. Otherwise **MVerify** returns 0.

Algorithm 2: The MSign algorithm in **IBMS$^{\mathbf{CR}}-2$**.

Input: the master public key mpk, the private key sk, the identity set $IDSet$, the message to be signed m;
Output: a multi-signature $m\sigma$.
1. Each MS_i randomly selects $r_i \in \mathbb{Z}_N^*$ and calculates $R_i = r_i^3 \ (mod\ N)$ and $t_i = h_2(R_i)$.
2. Each MS_i broadcasts t_i to co-signers MS_j $(j \neq i)$.
3. After obtaining t_i from MS_i, MS_1 broadcasts R_1 to other MS_i.
4. After receiving R_i from other signers, MS_1 checks whether $t_i = h_2(R_i)$ for $2 \leqslant i \leqslant n$ is satisfied.
5. If one of these fails, the algorithm stops, which means the attackers have mixed invalid partial signatures. Otherwise, MS_1 sets $R = \prod_{i=1}^n R_i \ (mod\ N)$, $w = h_3(R \| IDSet \| m)$, and $u_1 = r_1 \cdot sk_1^w \ (mod\ N)$.
6. S_1 broadcasts u_1 to other MS_i.
7. After receiving u_i from MS_i, MS_1 aggregates these by $u = \prod_{i=1}^n u_i \ (mod\ N)$.
8. Each MS_i locally generates a multi-signature $m\sigma = (w, u)$.
Return $m\sigma$;

5.2. Correctness

The correctness is as follows:

$$u^3 \cdot \prod_{i=1}^n I_i^w \equiv \prod_{i=1}^n u_i^3 I_i^w \equiv \prod_{i=1}^n r_i^3 \cdot (sk_i^3 \cdot I_i)^w \equiv \prod_{i=1}^n R_i \equiv R \pmod N$$

5.3. Security Proof

IBMS$^{\mathbf{CR}}-2$ is secure under the factorization in the random oracle model.

Theorem 7. *If integer factorization is (t', ϵ')-hard, our **IBMS$^{CR}-2$** scheme is $(t, q_H, q_E, q_S, n, \epsilon)$-secure against existential forgery in the random oracle model.*

Because most of the simulation game between $\mathcal{A}$ and $\mathcal{C}$ is the same, we give the security proof simply.

Proof. When it is given an integer factorization instance N, $\mathcal{C}$ returns p or q if $\mathcal{A}$ succeeds in forging a multi-signature.

$\mathcal{C}$ sends $mpk = \{N, h_1, h_2, h_3, a, \eta, \lambda\}$ to $\mathcal{A}$. $\mathcal{C}$ maintains several lists ($list_{h_1}$, $list_{h_2}$, $list_{h_3}$, $list_S$).

- h_1**-Query**. $\mathcal{C}$ manages a list (ID, c, h_1, s). $\mathcal{C}$ sends h_1 to $\mathcal{A}$ if ID exists when $\mathcal{A}$ queries the hash value of ID. Otherwise, $\mathcal{C}$ randomly selects $s \in \mathbb{Z}_N^*$ and $c \in \{0, 1, 2\}$, sets $h_1 \equiv s^3/a^c \pmod N$, returns h_1, and adds (ID, c, h_1, s) to $list_{h_1}$.
- The h_2**-query**, h_3**-query** and **extraction query** are similar to **IBMS$^{CR}-1$**.
- The **multi-signature query** is similar to **IBMS$^{CR}-1$**, except that Equation (5) changes to

$$R_1 = u_1^3 \prod_{i=1}^{n} \left(a^{c_{ID_i}} \cdot h_1(ID_1)\right)^{-w^*} \tag{14}$$

At the end of the game, $\mathcal{A}$ forges $m\sigma^* = (w^*, u^*)$ with $IDSet^*$ on m^*. $\mathcal{C}$ calculates

$$R^* \leftarrow (u^*)^3 \prod_{i=1}^{n} \left(a^{c_{ID_i^*}} \cdot h_1(ID_i^*)\right)^{-w^*} \tag{15}$$

and queries $h_3(R^* \| IDSet^* \| m^*)$ to the hash oracle. If the forgery is valid, we obtain that

$$u^{*3} \equiv R^* \prod_{i=1}^{n} \left(a^{c_{ID_i^*}} \cdot h_1(ID_i^*)\right)^{w^*} \equiv R^* \prod_{i=1}^{n} \left(s_i^{*3}\right)^{w^*} \equiv R^* s^{*w^*} \pmod N \tag{16}$$

because $s^* \leftarrow \prod_{i=1}^{n} (s_i^*)^3 \pmod N$. $\mathcal{C}$ returns (s^*, w^*, u^*).

We also apply the rewinding technique to factor N. At last, $\mathcal{C}$ obtains (s, w, u) and (s', w', u') such that

$$u^3 \equiv Rs^w \ \ and \ \ u'^3 \equiv R's'^{w'} \tag{17}$$

Because $R = R'$, $m = m'$, and $s = s'$, we have

$$\left(\frac{u}{u'}\right)^3 \equiv s^{(w-w')} \pmod N \tag{18}$$

Because $w \neq w'$, two cases emerge:

- If $w - w' \equiv 1 \pmod 3$, we denote $w - w' = 3k + 1$ for an integer k. Therefore, $s \equiv \left(\frac{u}{u' \cdot s^k}\right)^3$, that is, $\tilde{s} = \frac{u}{u' \cdot s^k}$ satisfies $\tilde{s}^3 \equiv s \pmod N$.
- If $w - w' \equiv -1 \pmod 3$, we denote $w - w' = 3k - 1$ for an integer k. Therefore, $s \equiv (\frac{u \cdot s^k}{u'})^3$, that is, $\tilde{s} = \frac{u \cdot s^k}{u'}$ satisfies $\tilde{s}^3 \equiv s \pmod N$.

From the discussion above, $\mathcal{C}$ calculates a cubic root $\tilde{s}$ where $\tilde{s}^3 = s$. Meanwhile $\mathcal{C}$ searches the entries in the h_1-list where $ID_i \in IDSet$ and calculates $\bar{s} = \prod_{i \in IDSet} s_i^3$. Therefore, we have $\tilde{s}^3 \equiv \bar{s}^3 \equiv s \pmod N$. If $\bar{s} \neq \tilde{s} \pmod N$, we can factor N by Theorem 1 with a probability that $\tilde{s} \neq \bar{s} \pmod N$ of 2/3.

Thus, we have finished the proof. □

6. Performance Comparisons

The comparison of security assumptions for related works are given in Table 1. These schemes are provably secure on the basis of different hardness assumptions (such as CDH, DL, RSA, quadratic residues, and cubic residues). The aim of these schemes is to find new constructions under simpler hardness assumptions.

Table 1. The comparison of related work on the security assumptions.

Schemes	The Underlying Mathematical Assumptions
[15]	Computational Diffie-Hellman (CDH)
[19]	Discrete Logarithm (DL)
[20]	RSA
[6]	Quadratic Residues
IBMS^{CR}-1	Cubic Residues
IBMS^{CR}-2	Cubic Residues

We denote M_p, H_m, O_p and E_n as the operation of scalar multiplication, map-to-point hash function, bilinear pairing, and modular exponentiation, respectively. We ran each of the above operations in a personal computer and used their times from [33] to calculate the total computational cost in the running time (milliseconds), as shown in the columns of Table 2.

Table 2. The comparison of related work of IBMS on the computational performance.

Schemes	Extract	Sign	Verify	Total Time	Length
[15]	$2H_m + 2M_p$	$1H_m + 4M_p$	$3O_p$	107.52	$2\,\|g\|$
[19]	$1E_n$	$2E_n$	$2E_n$	26.55	$\ell + \|N\|$
[20]	$1E_n$	$2E_n$	$2E_n$	26.55	$\ell + 2\|N\|$
[6]	$1E_n$	$2E_n$	$2E_n$	26.55	$\ell + \|N\|$
IBMS^{CR}-1	$1E_n$	$2E_n$	$2E_n$	26.55	$\ell + \|N\|$
IBMS^{CR}-2	$2E_n$	$1E_n$	$1E_n$	21.24	$\ell + \|N\|$

We have also compared related works on the basis of the cubic residues for the computational performance evaluation in Table 3. For consistency, we used the modular exponentiation times to evaluate the **Sign** and **Verify** algorithms.

Table 3. The comparison of related work on computational performance based on the cubic residues.

Schemes	Underlying Cryptographic Primitive	Sign	Verify	Total Time
[28]	IBMPS	$3E_n$	$3E_n$	$6E_n$
[26]	IBPMS	$1E_n$	$3E_n$	$4E_n$
[29]	IBMPMS	$3E_n$	$3E_n$	$6E_n$
IBMS^{CR}-1	IBMS	$2E_n$	$2E_n$	$4E_n$
IBMS^{CR}-2	IBMS	$1E_n$	$1E_n$	$2E_n$

7. Conclusions

Data authenticated aggregation is always a significant issue for marine WSNs. Most data authenticated aggregation is based on the multi-signature, which relies on the technique of bilinear pairing involving heavy computational overhead or the management of certificates beyond marine wireless sensors. We have constructed two efficient IBMS schemes (**IBMS^{CR}−1** and **IBMS^{CR}−2**) based on cubic residues, which are much more suitable for data authenticated aggregation in marine WSNs. Without employing the heavy overload of a bilinear pairing technique, our schemes have been

designed efficiently. Our schemes have been proven to be secure under chosen identity attacks and chosen message attacks, relying only on the hardness of the integer factorization assumptions.

Acknowledgments: This work was supported by the Natural Science Foundation of China (NSFC grant nos. 61402282, 61672339 and 41671431), the Shanghai Youth Talent Development Program (grant no. 14YF1410400), and the Shanghai Local University Capacity Enhancement Program (grant no. 15590501900).

Author Contributions: The work was conducted under the cooperation of authors. Lifei Wei conceived the scheme 1, and wrote the partial paper; Lei Zhang conceived the scheme 2 and wrote the partial paper, Dongmei Huang guided the study and reviewed the manuscript; Kai Zhang conceived the security proof and wrote the partial paper; Liang Dai gave the figures and verified the results; Guojian Wu introduced the background of marine sensor networks.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Bosman, H.; Iacca, G.; Tejada, A.; Wortche, H.J.; Liotta, A. Spatial anomaly detection in sensor networks using neighborhood information. *Inform. Fusion J.* **2017**, *33*, 41–56.
2. Bosman, H.; Iacca, G.; Tejada, A.; Wortche, H.J.; Liotta, A. Ensembles of incremental learners to detect anomalies in ad hoc sensor networks. *Ad Hoc Netw.* **2015**, *35*, 14–36.
3. Ahn, J.; Green, M.; Hohenberger, S. Synchronized aggregate signatures: New definitions, constructions and applications. In Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS 2010), Chicago, IL, USA, 4–8 October 2010.
4. Wei, L.; Zhang, L.; Huang, D.; Zhang, K. Efficient and Provably Secure Identity-based Multi-Signature Schemes for Data Aggregation in Marine Wireless Sensor Networks. In Proceedings of the 14th IEEE International Conference on Networking, Sensing and Control (ICNSC 2017), Calabria, Italy, 16–18 May 2017.
5. Huang, D.; Zhao, D.; Wei, L.; Wang, Z.; Du, Y. Modeling and analysis in marine big data: Advances and challenges. *Math. Probl. Eng.* **2015**, *2015*, 1–13.
6. Wei, L.; Cao, Z.; Dong, X. Secure identity-based multisignature schemes under quadratic residue assumptions. *Secur. Commun. Netw.* **2013**, *6*, 689–701.
7. Hsiao, H.; Studer, A.; Chen, C.; Perrig, A.; Bai, F.; Bellur, B.; Iyer, A. Flooding-resilient broadcast authentication for vanets. In Proceedings of the 17th Annual International Conference on Mobile Computing and Networking (MOBICOM 2011), Las Vegas, NV, USA, 20–22 September 2011.
8. Itakura, K.; Nakamura, K. A public-key cryptosystem suitable for digital multisignatures. *NEC Res. Dev.* **1983**, *71*, 1–8.
9. Barr, K.C.; Asanovic, K. Energy-aware lossless data compression. *ACM Trans. Comput. Syst.* **2006**, *24*, 250–291.
10. Bagherzandi, A.; Cheon, J.; Jarecki, S. Multisignatures secure under the discrete logarithm assumption and a generalized forking lemma. In Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS 2008), Alexandria, VA, USA, 27–31 October 2008.
11. Ma, C.; Weng, J.; Li, Y.; Deng, R. Efficient discrete logarithm based multi-signature scheme in the plain public key model. *Des. Codes Cryptogr.* **2010**, *54*, 121–133.
12. Shamir, A. Identity-based cryptosystems and signature schemes. In Proceedings of the 4th International Cryptology Conference (CRYPTO 1984), Santa Barbara, CA, USA, 19–22 August 1984.
13. Boneh, D.; Franklin, M. Identity-based encryption from the Weil pairing. *SIAM J. Comput.* **2003**, *32*, 586–615.
14. Cocks, C. An Identity Based Encryption Scheme Based on Quadratic Residues. In Proceedings of the 8th IMA International Conference on Cryptography and Coding, Cirencester, UK, 17–19 December 2001.
15. Gentry, C.; Ramzan, Z. Identity-based aggregate signatures. In Proceedings of the 9th International Conference on Theory and Practice of Public-Key Cryptography (PKC 2006), New York, NY, USA, 24–26 April 2006.
16. Lu, R.; Lin, X.; Zhu, H.; Liang, X.; Shen, X. BECAN: A Bandwidth-Efficient Cooperative Authentication Scheme for Filtering Injected False Data in Wireless Sensor Networks. *IEEE Trans. Parallel Distrib. Syst.* **2012**, *23*, 32–43.

17. Zhang, K.; Wei, L.; Li, X.; Qian, H. Provably Secure Dual-Mode Publicly Verifiable Computation Protocol in Marine Wireless Sensor Networks. In Proceedings of the 10th International Conference on Wireless Algorithms, Systems, and Applications (WASA 2017), Guilin, China, 19–21 June 2017.
18. Lu, Y.; Li, J. A Pairing-Free Certificate-Based Proxy Re-encryption Scheme for Secure Data Sharing in Public Clouds. *Future Gener. Comput. Syst.* **2016**, *62*, 140–147.
19. Bellare, M.; Neven, G. Identity-Based Multi-signatures from RSA. In Proceedings of the Cryptographers Track at the RSA Conference (CT-RSA 2007), San Francisco, CA, USA, 5–9 February 2007.
20. Bagherzandi, A.; Jarecki, S. Identity-Based Aggregate and Multi-Signature Schemes Based on RSA. In ,Proceedings of the 13th International Conference on Practice and Theory in Public Key Cryptography (PKC 2010), Paris, France, 26–28 May 2010; pp. 480–498.
21. Yang, F.; Lo, J.; Liao, C. Improving an efficient id-based rsa multisignature. *J. Ambient Intell. Hum. Comput.* **2011**, *4*, 249–254.
22. Chai, Z. Cao, Z.; Dong, X. Identity-based signature scheme based on quadratic residues. *Sci. China Inform. Sci.* **2007**, *50*, 373–380.
23. Xing, D.; Cao, Z.; Dong, X. Identity based signature scheme based on cubic residues. *Sci. China Inform. Sci.* **2011**, *54*, 2001–2012.
24. Wang, Z.; Wang, L.; Zheng, S.; Yang, Y.; Hu, Z. Provably secure and efficient identity-based signature scheme based on cubic residues. *Int. J. Netw. Secur.* **2012**, *14*, 33–38.
25. Wang, F.; Lin, C. Secure and efficient identity-based proxy multisignature using cubic residues. *J. Univ. Electr. Sci. Technol. China* **2013**, *42*, 778–783.
26. Wang, F.; Chang, C.-C.; Lin, C.; Chang, S.-C. Secure and efficient identity-based proxy multi-signature using cubic residues. *Int. J. Netw. Secur.* **2016**, *18*, 90–98.
27. Wang, F.; Lin, C.; Lian, G. Efficient identtiy based threshold ring signature based on cubic residues. *J. Wuhan Univ. (Nat. Sci.)* **2013**, *59*, 75–81.
28. Wei, L.; Zhang, L.; Zhang, K.; Dong, M. An Efficient and Secure Delegated Multi-Authentication Protocol for Mobile Data Owners in Cloud. In Proceedings of the 10th International Conference on Wireless Algorithms, Systems, and Applications (WASA15), Qufu, China, 10–12 August 2015.
29. Zhang, L.; Wei, L.; Huang, D.; Zhang, K.; Dong, M.; Ota, K. Medaps: Secure multi-entities delegated authentication protocols for mobile cloud computing. *Secur. Commun. Netw.* **2016**, *9*, 3777–3789.
30. Damgard, I.; Frandsen, G. Efficient algorithms for gcd and cubic residuosity in the ring of Eisenstein integers. *J. Symb. Comput.* **2005**, *39*, 643–652.
31. Benhamouda, F.; Herranz, J.; Joye, M.; Libert, B. Efficient cryptosystems from 2^k. *J. Cryptol.* **2016**, 1–31.
32. Coron, J. On the exact security of full domain hash. In Proceedings of the 20th Annual International Cryptology Conference (CRYPTO 2000), Santa Barbara, CA, USA, 20–24 August 2000.
33. He, D.; Chen, J.; Zhang, R. An efficient and provably-secure certificateless signature scheme without bilinear pairings. *Int. J. Commun. Syst.* **2012**, *25*, 1432–1442.