Hindawi Publishing Corporation
Mobile Information Systems
Volume 2015, Article ID 827320, 11 pages
http://dx.doi.org/10.1155/2015/827320

Research Article

Attribute Based Multisignature Scheme for Wireless Communications

Ximeng Liu,[1] Hui Zhu,[2] Jianfeng Ma,[3] Qi Li,[3] and Jinbo Xiong[4]

[1] *School of Telecommunications Engineering, Xidian University, Xi'an, Shaanxi 710071, China*
[2] *School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798*
[3] *School of Computer Science and Technology, Xidian University, Xi'an, Shaanxi 710071, China*
[4] *Faculty of Software, Fujian Normal University, Fuzhou, Fujian 350108, China*

Correspondence should be addressed to Ximeng Liu; snbnix@gmail.com

Received 29 August 2014; Accepted 1 September 2014

Academic Editor: David Taniar

With rapidly development of wireless communication, more mobile devices are used in our daily life. Although the need for accessing a wireless network is evident, new problems, such as keeping and preserving user identity's privacy, should be greatly concerned. Attribute based signature scheme is an important cryptographic primitive which provides a powerful way for user to control their privacy. In wireless environment, the capacity of wireless channel is also valuable resources which is limited. More information can be transmitted through the wireless channel when the cost of using signature to verify the message becomes less. In order to reduce the bandwidth needed to transmit attribute based signatures and keep signer's privacy, attribute based multisignature scheme (ABMS) was proposed in this paper. Moreover, we formalize and construct the ABMS. Our scheme is existentially unforgeable against chosen message attack on Computational Diffie-Hellman (CDH) assumption in the standard model. The simulation shows that our ABMS scheme is more appropriate for wireless communication to guarantee integrity of the data.

1. Introduction

With the increasing availability of mobile devices, it is convenient for people to make a phone call and surf the internet through the wireless channel. With features of convenient, fast, and easy-to-use, there is a growing demand for consumer to transmit data through the wireless channel. Due to the character of the wireless channel, the data can be easily changed which is affected by transmission channel noise or modified by the malicious attacker. The security and privacy protection of the data collected from wireless devices, either while stored in the data server or during their transmission through the wireless network, is a major concern. Also, preserving identity privacy becomes an increasingly important concern. In Oct. 2013, the attackers are believed to have stolen information on 2.9 million Adobe account holders. That data includes customer names, encrypted credit and debit card numbers, expiration dates, and other customer order information [1]. How to efficiently verify the data integrity and preserve identity privacy is important problem in the wireless environment.

Attribute based signatures (ABS) [2] scheme has attracted much attention as a new public key primitive in the recent years because it provides a powerful way for user to control their privacy and keep the integrity of the data, and it also helps to provide fine-grained access control in anonymous authentication systems. The ABS scheme is analogue of attribute based encryption (ABE) [3, 4] which is an important application of the fuzzy identity-based encryption (FIBE) scheme [3]. A user encrypts a message with a set of n attributes such that users whose decryption key has at least t common attributes with the ciphertext attribute set can decrypt the message. We call this scheme threshold attribute based encryption (t-ABE) to describe simplicity. Wang et al. [5] proposed a new fully secure FIBE scheme based on the FIBE [3] scheme and prove its security by using the "dual system encryption" technique. The ABS scheme extends identity-based signature where the signer is associated with a set of attributes instead of a single identity string. It provides a powerful way for users to control their privacy: the user can choose the subset of their attributes relevant to the specific scenario in signing a document. Considering the following scenario, an institution will release a technical report that may involve a professor at age 45 in the computer science department. Any user who has attributes sets that contain all the above attributes could issue the signature. Because ABS scheme has these advantages, different user wants to sign the same document by using ABS scheme. Yang et al. [6] introduced a new cryptographic primitive called fuzzy identity-based signature (FIBS) which the signature analogue of FIBE scheme and Shahandashti and Safavi-Naini [7] proposed a threshold attribute based signature construction for small attribute universe and large attribute universe. Since FIBS scheme lacks controlling the signer's privacy, Maji et al. [8] introduced ABE scheme which can provide strong privacy guarantee for the signer and strong unforgeability guarantee for the verifier. In order to sign messages with any subset of their attributes issued from an attribute center, Li and Kim [9] gave hidden attribute based signatures without anonymity revocation scheme which can reach anonymity and unforgeability. Li et al. [10] proposed a new construction of ABS supporting flexible threshold predicate which could compact the signature size and improve the verification time. Later, Cao et al. [11] give multiauthority attribute based signature schemes for expressive policy. In their scheme, they use both AND, OR, threshold, and disjunctive normal form to express a policy. Consider the following case; users often use wireless channel to upload file to the data center. Unfortunately, these communication mechanisms are rather expensive for mobile devices in energy consumption and the capacity of wireless channel is limited. In order to increase throughput of message sent to the data center and increase the battery life of the energy-restricted devices, it is better to exploit fewer bits of transmission in wireless communication to data center. Therefore, it is a challenge to design cryptographic primitives to reduce the communication and storage overhead.

Multisignatures allow multiple signers to jointly authenticate a message using a single compact signature which was first introduced by [12]. It allows a group of players to sign the same message by generating a short signature which can be verified against the set of these players' public keys. After that, lots of multisignature schemes were proposed in [13–15]. But these schemes lacked formal security notions for multisignatures. Micali et al. [16] first formalized the strong notion of security for multisignatures and [17] gave a more general construct in random oracle model where their construction did not restrict the subset of signers. The security is based on random oracles. Lu et al. [18] first proposed sequential aggregate signature and multisignature scheme in the standard model. Because the verification information of identity-based signature (IBS) scheme does not include any certificate or any individual public key for the signer, identity-based multisignature (IDMS) scheme was presented by Cheon et al. [19]. This scheme could reduce the signature size into almost a half and efficiently verify multiple signatures. Gentry and Ramzan [20] designed the efficient identity-based (Multi-/Aggregate) signatures. Their schemes employ a group with a bilinear map in the random oracle model. Later, there are several RSA-based IBMS schemes proposed whose security is based on RSA assumption. The computational costs of RSA-based IBMS scheme are slightly lower in signing and verification because RSA exponentiation is less expensive than bilinear map operations. Recently, Liu et al. [21] proposed an attribute based multisignature scheme in the standard model with can reduce the length of signature. However, the performance of this ABMS scheme is not good. Later, Liu et al. [22] proposed another ABMS scheme for the wireless environment. But the authors do not give performance measurement to show their scheme is efficient.

In this paper, we first propose a scheme called attribute based multisignature (ABMS) scheme to solve problem mentioned above. The ABMS scheme allows a set of signatures (sign on the same message) to be compressed into a single signature. This kind of signature has less signature length than the original one and less computational cost which is more appropriate for the wireless nature where bandwidth is a bottleneck.

Our Contributions. In this work, we make following contributions: (1) We define attribute based multisignature scheme (ABMS), formalize the model, and give security model for ABMS scheme. (2) We give overview of ABMS scheme for wireless communication and a concrete construction of ABMS scheme. (3) We prove that our ABMS scheme is existential unforgeability in the standard model by using the computational Diffie-Hellman problem. (4) We make simulation on a workstation to show that ABMS scheme can greatly decrease the storage overhead in the data center and computational overhead for verifier.

Organization. The rest of paper is organized as follows. In Section 2, we review some concept about bilinear pairing, complexity assumption, flexible threshold predicate, and Lagrange interpolation. In Section 3, we give the formal models and its security model of ABMS scheme. In Section 4, we give the specific construction about the ABMS scheme. In Section 5, we give security proof in the standard model for ABMS scheme. In Section 6, we give performance analysis on ABMS scheme, use the workstation to test the performance of ABMS scheme, and analyze the efficiency of the ABMS scheme. And we conclude this paper in Section 7.

2. Preliminaries

In this section, we introduce the notions which are used to construct ABMS scheme and prove the security of ABMS scheme.

2.1. Bilinear Maps. Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic groups of prime order p with the multiplication. Let g be a generator of $\mathbb{G}$ and e a bilinear map. Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map having the following properties:

(1) bilinearity: For all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$;

(2) nondegeneracy: $e(g, g) \neq 1$;

(3) computability: There is efficient algorithm to compute bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$.

Notice that the map e is symmetric since $e(u^a, v^b) = e(u, v)^{ab} = e(u^b, v^a)$.

2.2. Complexity Assumptions

Definition 1. The challenger chooses $a, b \in \mathbb{Z}_p$ at random and outputs (g, g^a, g^b). The computational Diffie-Hellman (CDH) problem is to compute g^{ab}. An adversary $\mathscr{A}$ has at least an ϵ if

$$\left|\Pr\left[\mathscr{A}\left(g, g^a, g^b\right) = g^{ab}\right]\right| \geq \epsilon. \quad (1)$$

The computational (t, ϵ)-DH assumption holds if no t-time adversary has at least ϵ advantage in solving the above game.

2.3. Flexible Threshold Predicate. In this paper, we use predicates Υ consisting of thresholds gates. All predicates $\Upsilon_{k,\omega^*}(\cdot) \rightarrow 0/1$ for ω^* with threshold value k. If the number of attribute in $\omega' \cap \omega^*$ exceeds threshold k, it outputs 1. Otherwise, it outputs 0. Consider

$$\Upsilon_{k,\omega^*}\left(\omega'\right) = \begin{cases} 1, & \left|\omega' \cap \omega^*\right| \geq k, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

2.4. Lagrange Interpolation. In this section, we describe Lagrange interpolation which is used in the ABMS schemes. Given d points $q(1), \ldots, q(d)$ on a $d-1$ degree polynomial, we can use Lagrange interpolation to compute $q(i)$ for any $i \in \mathbb{Z}_p$. Let S be a d-element set. We define the Lagrange coefficient $\Delta_{j,S}(i)$ of $q(j)$ in the computation of $q(i)$ as

$$\Delta_{j,S}(i) = \prod_{\eta \in S, \eta \neq j} \frac{i - \eta}{j - \eta}. \quad (3)$$

2.5. Symbols & Notations. The following list shows the symbols and notations used in this work:

$e(\cdot, \cdot)$: bilinear maps,

k: threshold gate,

$\Upsilon_{k,\omega^*}(\cdot)$: predicates consisting of threshold gate k,

$\Delta_{j,S}$: Lagrange coefficient with set S,

params: public parameters,

D: private key,

σ: original signature,

$p\sigma_p$: multisignature,

$\omega, \omega', \omega^*, \widehat{\omega}, \Omega, \Omega'$: attribute set.

3. Formal Models and Its Security Model

3.1. Formal Models of ABMS Scheme. The attribute based multisignature scheme has six algorithms called *Setup*, *Extract*, *StandardSign*, *StandardVerify*, *MComb*, and *MultiVerify*. In this section, we describe the six algorithms as follows.

Setup. This algorithm is run by the master entity which inputs the security parameter and generates the public parameters *params* of the scheme and the master secret key MSK. The master entity publishes *params* and keeps the MSK to itself.

Extract. Given an attribute set ω, the master key MSK and *params*, the master entity will use this algorithm to generate private keys of ω for all entities participating in the scheme and distribute the private keys to their respective owner through a secure channel.

StandardSign. Given a message m, an attribute set ω, a private key D, *params*, and predicate $\Upsilon_{\omega^*}(\cdot)$, this algorithm generates the signature σ of ω on m. The entity with attribute set ω will use this algorithm for signing.

StandardVerify. Given a signature σ, a message m, attribute set ω, and *params*, this algorithm outputs *accept* if a valid signature on message for attribute set and outputs *reject* otherwise.

MComb. The algorithm is given a signature-public key pair $\{(\sigma_i, params_i)\}_{i=1}^{l}$ and a message m. The l is the number of user's signing the message m. It generates and outputs a multisignature $p\sigma_p$.

MultiVerify. The algorithm is given the public parameters $\{param_i\}_{i=1}^{l}$, a message m, and multisignature $p\sigma_p$. The algorithm outputs *accept* if it is a valid multisignature and outputs *reject* otherwise.

3.2. Existential Unforgeability of ABMS Scheme. We define security model for attribute based multisignature scheme between a challenger and an adversary.

Setup. The challenger runs the *Setup* algorithm and obtains both the public parameters *params* and the master secret key. The challenger gives the *params* to adversary and keeps the master secret key by itself.

Queries. The adversary adaptively makes a polynomial bounded number of queries to the challenger. Each query can be one of the following.

(i) *Extract Query*. The adversary can ask for the private key of any attribute set ω. The challenger responds by running the *Extract* algorithm and gives the private key to adversary.

(ii) *Sign Query*. The adversary can ask for the signature of attribute set ω on message m. The challenger responds by first running *Extract* algorithm to obtain the private key and running the *Sign* algorithm to obtain a signature which is given to the adversary.

Output. Eventually, it will output a forgery σ^* on messages m under public parameters $\{params_i\}_{i=1}^{l}$. The challenger key

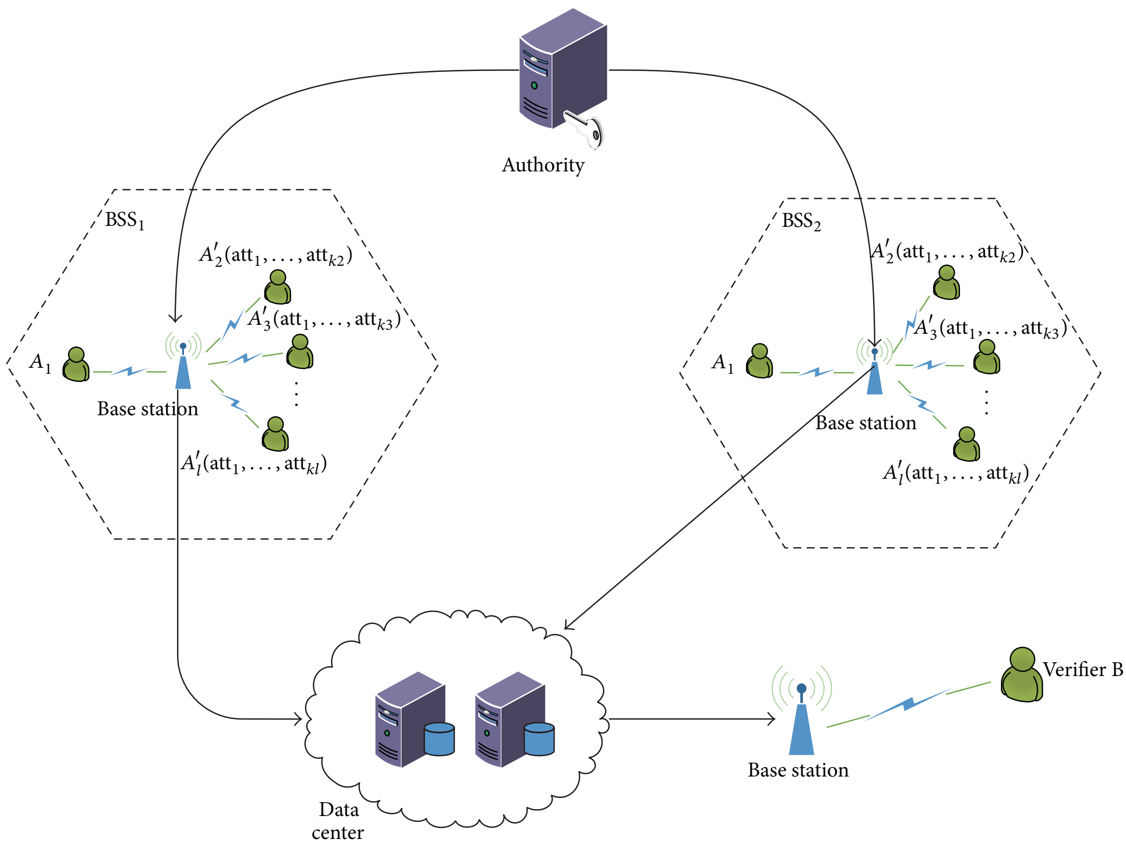


Figure 1: ABMS scheme for wireless network.

must appear in $\{params_i\}_{i=1}^{l}$, without loss of generality; we assume that the challenge key appears at index 1. If the condition holds, it outputs 1. Otherwise, it outputs 0.

Definition 2. The attribute based multisignature scheme is (t, q_e, q_S, ϵ)-secure against existential forgery in an adaptive chosen-message attack, if no t-time adversary makes q_e *Extract queries*, q_S *Sign queries* and wins the above game with advantage more than ϵ.

4. Our Constructions

In this section, we first give the overview of the whole wireless communication system and then give a concrete construction of the ABMS scheme.

4.1. Overview of Privacy-Preserving Data Integrity Verification Method for Wireless Communication. Bandwidth is scarce resources in the wireless communication. In order to verify the data integrity, the signature method will be brought into the system. But it will greatly increase the communication cost especially when the number of users involved in the system is huge. Meanwhile, the mobile devices are always energy-restricted, such as mobile phone and wireless sensor nodes. More extra computation will increase the consumption of battery power. The main goal of our attribute based multisignature scheme is to reduce both communication overhead and verification cost in order to keep data integrity in the process of wireless communication. Also, it could allow user to control their identity's privacy. The whole system model can be showed in Figure 1. As Figure 1 shows, there are message provider (A_1), a group of signers $(A'_2, A'_3, \ldots, A'_l)$, verifier B, and authority involved in the system. The authority first generates the master key and defines a common universe of attributes, such as "headmaster," "professor," "age 45," and "computer science department." Then the authority uses master key and attribute sets to construct $A_1, A'_2, \ldots, A'_l$'s private key and send it to the corresponding users involved in the system, respectively. Because the message needs to be signed by message provider and a group of signers, the provider first generates the message and the signature associated with the message and then sends it to the group of users. All the users in the same BSS need to sign the message. When signers $(A'_2, A'_3, \ldots, A'_l)$ receive the message-signature pair $(m, sign_m)$, they should first verify whether or not the $(m, sign_m)$ is sent by message provider A_1. If $(m, sign_m)$ passed the verification, it is considered that the message is sent by A_1 and used m to generate his own message and signature pair $(m, sign_{A'_k})$, $k = 2$ to l. Then the message and signature pair $(m, sign_m)$ and $(m, sign_{A'_k})$, $k = 2$ to l should be compressed into a single message-multisignature pair $(m, sign_{\mathrm{MS}})$ and it is sent to data center to store. When another user B needs to use message m, she/he first retrieves the message from the data center and uses signature $sign_{\mathrm{MS}}$

to verify the message. If the verification holds, we say this message is integrated which is signed by the user A'_k ($k = 2$ to l). Otherwise, it shows that the message is modified by the third party servers. If we use traditional methods, we need to transmit l pairs to the verifier. When we use ABMS scheme, the only thing is to create one message-signature pair to transmit in the network which can greatly decrease the transmitting overhead through the network and reduce the storage cost in the data center. The concrete construction of ABMS scheme will be presented in the next section.

4.2. Attribute Based Multisignature Scheme. In this section, we give a concrete construction of the ABMS scheme which contains six algorithms: *Setup*, *Extract*, *StandardSign*, *StandardVerify*, *MComb*, *MultiVerify*.

Setup. This algorithm first defines the attributes in the universe $\mathscr{U}$ as the element in $\mathbb{Z}_p$. A $d-1$ default attribute set from $\mathbb{Z}_p$ is given as $\Omega = \{\Omega_1, \Omega_2, \ldots, \Omega_{d-1}\}$. It selects a random generator $g \in \mathbb{G}$ and a random $\alpha_i \in \mathbb{Z}_p^*$ and compute $g_{1i} = g^{\alpha_i} \in \mathbb{G}$. Next, it picks a random element g_2 and computes $A^{(i)} = e(g_{1i}, g_2)$. For every user i, select a random vector $\mathbf{t} = (t_1, t_2, \ldots, t_{l+d-1})$ from $\mathbb{Z}_p^{l+d-1}$ and then compute $\mathbf{T}_i = (T_1 = g^{t_1}, T_2 = g^{t_2}, \ldots, T_{l+d-1} = g^{t_{l+d-1}})$. Finally, the algorithm selects random values y' from $\mathbb{Z}_p$ and a random vector $\mathbf{y} = (y_1, y_2, \ldots, y_k)$ from $\mathbb{Z}_p^k$ and computes $\mathbf{U} = (u_1, u_2, \ldots, u_k) = (g^{y_1}, g^{y_2}, \ldots, g^{y_k})$. The public parameters are

$$params = \left(\mathbb{G}, \mathbb{G}_T, e, g, g_1, g_2, \mathbf{T}, \mathbf{U}\right). \tag{4}$$

Here, for different users, the public keys are denoted as

$$\mathrm{PK}^{(i)} = A^{(i)}. \tag{5}$$

The master keys are

$$\mathrm{MSK}^{(i)} = \alpha_i. \tag{6}$$

Extract. This algorithm generates a private key for an attribute set ω related with users involved in the system. It takes the following steps.

(1) Firstly, it chooses a $d-1$ degree polynomial at random with $q(0) = \alpha_i$.

(2) It then generates a new attribute set $\widehat{\omega} = \omega \cup \Omega$. For each $i \in \widehat{\omega}$, the algorithm chooses and computes $d_{i0} = g_2^{q(0)} \cdot (g_1 T_i)^{r_i}$, $d_{i1} = g^{r_i}$.

(3) Finally, it outputs

$$D = \left(d_{i0}, d_{i1}\right)_{i \in \widehat{\omega}} \tag{7}$$

as the private key.

StandardSign. This algorithm inputs a private key for the attribute set ω, message m, and predicate $\Upsilon_{u,\omega^*}(\cdot)$. In order to sign message m with predicate $\Upsilon_{u,\omega^*}(\cdot)$, that is, to prove the signer owning at least u attribute among the c-elements attribute set ω^*. It selects a u-element from the subset $\omega' \subseteq \omega \cap \omega^*$ and works as follows.

(1) First, it selects a default attribute subset $\Omega' \subseteq \Omega$ with $|\Omega'| = d - u$ and chooses $c + d - u$ random values $r'_i \in \mathbb{Z}_p$ for $i \in \omega^* \cup \Omega'$.

(2) It then computes

$$\sigma_0 = \left[\prod_{i \in \omega' \cup \Omega'} d_{i0}^{\Delta_{i,S}(0)}\right] \left[\prod_{i \in \omega^* \cup \Omega'} \left(g_1 T_i\right)^{r'_i}\right] \left(u' \prod_{j \in \mathscr{M}} u_j^{m_j}\right)^{r_s},$$
$$\left\{\sigma_i = d_{i1}^{\Delta_{i,S}(0)} g^{r'_i}\right\}_{i \in \omega' \cup \Omega'}, \qquad \left\{\sigma_i = g^{r'_i}\right\}_{i \in \omega^* / \omega'},$$
$$\sigma'_0 = g^{r_s}. \tag{8}$$

(3) Finally, the algorithm outputs the signature:

$$\sigma = \left(\sigma_0, \{\sigma_i\}_{i \in \omega^* \cup \Omega'}, \sigma'_0\right). \tag{9}$$

StandardVerify. In order to verify the correctness of the signature $\sigma = (\sigma_0, \{\sigma_i\}_{i \in \omega^* \cup \Omega'}, \sigma'_0)$ on m with threshold k for attributes set $\omega^* \cup \Omega'$, it checks if the following equation holds:

$$\frac{e\left(g, \sigma_0\right)}{\left[\prod_{i \in \omega^* \cup \Omega'} e\left(g_1 T_i, \sigma_i\right)\right] e\left(u' \prod_{j \in \mathscr{M}} u_j^{m_j}, \sigma'_0\right)} = A^{(i)}. \tag{10}$$

If the equation holds, it indicates that the signature is indeed from some users with k attributes among ω^*. Otherwise, it denotes the signature is not valid.

MComb. For each user in the multisignature, the algorithm inputs a public parameters *params*, public key $\mathrm{PK}^{(k)}$, and a signature $\sigma^{(k)}$. All the signatures are signed on a single message m. Let m be an m'-bit message to be signed by the original signers $A'_1, A'_2, \ldots, A'_l$ and m_d denote the dth bit of m, and let $\mathscr{M} \subseteq \{1, 2, \ldots, m'\}$ be the set of all d for which $m_d = 1$. Denote $\mathrm{PK}^{(k)}$ as user k's public keys and its corresponding signature σ_k as $\sigma^{(k)} = (\sigma_0^{(k)}, \{\sigma_i\}_{i \in \omega^* \cup \Omega'}^{(k)}, \sigma_0'^{(k)})$. Verify that $\sigma^{(k)}$ is valid by calling the *StandardVerify* algorithm. If not, its outputs fail and halt. Otherwise, the algorithm takes following steps.

For each user in the multisignature the algorithm inputs a public parameters *params*, public key $\mathrm{PK}^{(k)}$, and a signature $\sigma^{(k)}$. All the signatures are signed on a single message m. Let m be an m'-bit message to be signed by the original signers $A'_1, A'_2, \ldots, A'_l$ and m_d denote the dth bit of m, and let $\mathscr{M} \subseteq \{1, 2, \ldots, m'\}$ be the set of all d for which $m_d = 1$. Denote $\mathrm{PK}^{(k)}$ as user k's public keys and its corresponding signature σ_k as $\sigma^{(k)} = (\sigma_0^{(k)}, \{\sigma_i\}_{i \in \omega^* \cup \Omega'}^{(k)}, \sigma_0'^{(k)})$. Verify that $\sigma^{(k)}$ is valid by calling the *StandardVerify* algorithm. If not, its outputs fail and halt. Otherwise, the algorithm takes following steps.

This algorithm first initializes $p\sigma_p$, and sets $p\sigma_{p0} = 1$ and $p\sigma_{p2} = 1$. For every i belong to $\omega' \cup \Omega'$ and ω^*/ω', sets $p\sigma_i = 1$. Also, the algorithm initializes $\omega' \cup \Omega' = \emptyset$ and $\omega^*/\omega' = \emptyset$.

For $k = 1, \ldots, l$, it calculates

$$p\sigma_{p0} = p\sigma_{p0} \cdot \sigma_0^{(k)},$$
$$p\sigma_{p2} = p\sigma_{p2} \cdot \sigma_0'^{(k)}. \tag{11}$$

Then for every $i \in \omega_k' \cup \Omega_k'$, if i does not exist in $\omega' \cup \Omega'$, it adds attribute i to the attribute set $\omega' \cup \Omega'$ and sets $p\sigma_i = d_{i1}^{\Delta_{i,S_k}(0)} g^{r_i'}$. If i exists in $\omega' \cup \Omega'$, it sets $\sigma_i^{(k)} = d_{i1}^{\Delta_{i,S_k}(0)} g^{r_i'}$ and calculates $p\sigma_i = p\sigma_i \cdot \sigma_i^{(k)}$.

For every $i \in \omega_k^*/\omega_k'$, if i does not exist in ω^*/ω', it adds attribute i to the attribute set ω^*/ω' and sets $p\sigma_i = g^{r_i'}$. If i exists in ω^*/ω', it sets $\sigma_i^{(k)} = g^{r_i'}$ and computes $p\sigma_i = p\sigma_i \cdot \sigma_i^{(k)}$.

The algorithm finally computes:

$$p\sigma_p = \left(p\sigma_{p0}, \{p\sigma_i\}_{i\in\omega'\cup\Omega'}, \{p\sigma_i\}_{i\in\omega^*/\omega'}, p\sigma_{p2}\right). \tag{12}$$

MultiVerify. Given the public parameters, public keys, a message $m \in \{0,1\}^{m'}$, and a signature $p\sigma_p = (p\sigma_{p0}, \{p\sigma_i\}_{i\in\omega'\cup\Omega'}, \{p\sigma_i\}_{i\in\omega^*/\omega'}, p\sigma_{p2})$, a verifier *accept* $p\sigma_p$ if the following equality holds:

$$\frac{e\left(p\sigma_{p0}, g\right)}{\left[\prod_{i\in\omega^*\cup\Omega'} e\left(g_1 T_i, p\sigma_i\right)\right] e\left(u' \prod_{j\in\mathcal{M}} u_j^{m_j}, p\sigma_{p2}\right)} = \prod_{i=1}^{l} A^{(i)}. \tag{13}$$

Otherwise, it outputs *reject*.

5. Security of ABMS Scheme

In this section, we first show the correctness of our ABMS scheme. Then we prove that our ABMS scheme is existential unforgeability by using hard problem introduced in Section 2.2.

5.1. Correctness. The signature $p\sigma_p$ generated from *MComb* algorithm can be easily checked by verifier:

$$\begin{aligned}
&\frac{e\left(p\sigma_{p0}, g\right)}{\left[\prod_{i\in\omega^*\cup\Omega'} e\left(g_1 T_i, p\sigma_i\right)\right] e\left(u' \prod_{j\in\mathcal{M}} u_j^{m_j}, p\sigma_{p2}\right)} \\
&= e\left(\prod_{k=1}^{l}\left(\left[\prod_{i\in\omega_k'\cup\Omega_k'} d_{i0}^{\Delta_{i,S_k}(0)}\right]\left[\prod_{i\in\omega_k^*\cup\Omega_k'} \left(g_1 T_i\right)^{r_{i,k}'}\right]\right.\right. \\
&\qquad \left.\left. \cdot \left(u' \prod_{j\in\mathcal{M}} u_j^{m_j}\right)^{r_{s,k}}\right), g\right) \\
&\quad \cdot \left(\left[\prod_{i\in\omega^*\cup\Omega'} e\left(g_1 T_i, p\sigma_i\right)\right] e\left(u' \prod_{j\in\mathcal{M}} u_j^{m_j}, \prod_{k=1}^{l} g^{r_{s,k}}\right)\right)^{-1} \\
&= \prod_{k=1}^{l} e\left(\prod_{i\in\omega_k'\cup\Omega_k'} g_2^{q(0)\Delta_{i,S_k}(0)}, g\right) \\
&= \prod_{k=1}^{l} e\left(g_2^{a_i}, g\right) = \prod_{i=1}^{l} A^{(i)}.
\end{aligned} \tag{14}$$

5.2. Existential Unforgeability. In this section, we show our ABMS scheme which is existential unforgeability by giving the following theorem.

Theorem 3. *The attribute based multisignature scheme is (t, q_e, q_S, ϵ)-unforgeable if the (t', ϵ')-CDH assumption holds where*

$$\begin{aligned}
\epsilon' &\geq \frac{\epsilon}{4\left(\binom{d-1}{d-k}\right)\left(n_m+1\right)\left(q_S\right)}, \\
t' &= t + \mathcal{O}\left(d\left(q_e+q_S\right) + n_m \cdot q_S\right)\rho \\
&\quad + \left(d\left(q_e+q_S\right) + n_m \cdot q_S\right)\tau,
\end{aligned} \tag{15}$$

and ρ and τ are the time for a multiplication and an exponentiation in $\mathbb{G}$, respectively.

Proof. We will assume that adversary $\mathcal{A}$ has advantage ϵ in attacking the scheme. We will construct the algorithm $\mathcal{B}$ that solve the CDH with probability at least ϵ'. The algorithm $\mathcal{B}$ will be given a group $\mathbb{G}$, a generator g, and the elements g^a and g^b. In order to use $\mathcal{A}$ to compute the g^{ab}, $\mathcal{B}$ must simulate a challenger for $\mathcal{A}$. Such a simulation can be created in the following way.

Setup. We assume $l_m = 2q_S$. Let the default attribute set be $\Omega = \{\Omega_1, \Omega_2, \ldots, \Omega_{d-1}\}$ for some predefined integer d. $\mathcal{B}$ first define h_i as $h_i = g^{t_i}$ where t_i is randomly chosen from $\mathbb{Z}_p$. Then it chooses a random $k \in \{0, \ldots, q\}$, and random numbers $x', x_1, \ldots, x_m \in \mathbb{Z}_{l_m}$. It also chooses additional random exponents $z', z_1, \ldots, z_m \in \mathbb{Z}_p$. Consider

$$u' = g_2^{x'-l_1 k_1'} g^{z'}, \qquad u_k = g_2^{x_j} g^{z_j}, \quad 1 \leq j \leq m. \tag{16}$$

To make the notion easy to follow, we define two functions $F(m)$, $J(m)$

$$\begin{aligned}
F(m) &= x' - k_1' l_1 - \sum_{j\in\mathcal{M}} x_j m_j, \\
J(m) &= z' + \sum_{j\in\mathcal{M}} z_j m_j.
\end{aligned} \tag{17}$$

The master secret key will be $g_2^{\alpha} = g_2^{a} = g^{ab}$ and the following equations holds:

$$u' \prod_{j\in\mathcal{M}} u_j^{m_j} = g_2^{F(m)} g^{J(m)}. \tag{18}$$

Extract Query. $\mathcal{A}$ can make requests for private key on Ω such that $|\omega \cap \omega^*| < k$. We first define three subsets Γ, Γ', S in the following manner: $\Gamma = (\omega \cap \omega^*) \cup \Omega^*$ and $\Gamma \subseteq \Gamma' \subseteq S$ and $|\Gamma'| = d-1$. Let $S = \Gamma' \cup \{0\}$. For $i \in \Gamma'$, compute $D_i = (g_2^{\lambda_i}(g_2 g^{t_i})^{r_i}, g^{r_i})$ where λ_i, r_i are randomly chosen in $\mathbb{Z}_p$. For $i \notin \Gamma'$, it could also simulate as

$$\begin{aligned}
D_1^{(i)} &= \left(\prod_{j\in\Gamma'} g_2^{\lambda_j \Delta_{j,S}(i)}\right)\left(g_1^{-t_i}\left(g_2 g^{t_i}\right)^{r_i'}\right)^{\Delta_{0,S}(i)} \\
&= \left(\prod_{j\in\Gamma'} g_2^{\lambda_j \Delta_{j,S}(i)}\right)\left(g^{-at_i}\left(g_2 g^{t_i}\right)^{r_i'}\right)^{\Delta_{0,S}(i)} \\
&= \left(\prod_{j\in\Gamma'} g_2^{\lambda_j \Delta_{j,S}(i)}\right)\left(g_2^{a}\left(g_2 g^{t_i}\right)^{-a}\left(g_2 g^{t_i}\right)^{r_i'}\right)^{\Delta_{0,S}(i)} \\
&= \left(\prod_{j\in\Gamma'} g_2^{\lambda_j \Delta_{j,S}(i)}\right)\left(g_2^{a}\left(g_2 g^{t_i}\right)^{r_i'-a}\right)^{\Delta_{0,S}(i)} \\
&= \left(\prod_{j\in\Gamma'} g_2^{\lambda_j \Delta_{j,S}(i)}\right) g_2^{a\Delta_{0,S}(i)}\left(g_2 g^{t_i}\right)^{r_i} \\
&= g_2^{q(i)}\left(g_2 T_i\right)^{r_i}, \\
D_2^{(i)} &= \left(g_1^{-1} g^{r_i'}\right)^{\Delta_{0,S}(i)} = \left(g^{r_i'-a}\right)^{\Delta_{0,S}(i)}.
\end{aligned} \tag{19}$$

Sign Query. Consider the query for a signature of attribute set on m. If $F(m) = 0(\bmod p)$, the simulation aborts. Otherwise, $\mathscr{B}$ selects a random set Λ such that and $|\Lambda| = d-1$. Define $g^{q'(i)} = g^{\lambda_i'}$ where λ_i' is chosen randomly in $\mathbb{Z}_p$. Then it computes $g^{q'(i)} = (\prod_{k=1}^{d-1} g^{\lambda_k' \Delta_{k,\omega_u^*}(i)}) g^{a\Delta_{0,\omega_u^*}(i)}$ for $i \in \omega^* - \Lambda$. $\mathscr{B}$ randomly picks $r_i', \widehat{r}_s \in \mathbb{Z}_p$ and computes the signature as

$$\sigma = \left(S_1, \left\{\sigma_{k_1}\right\}_{i\in\omega}, S_3\right), \tag{20}$$

where

$$\begin{aligned}
S_1 &= g_1^{-J(m)/F(m)}\left[\prod_{i\in\omega'\cup\Omega'}\left(g_2 T_i\right)^{r_i\Delta_{i,S}(0)}\right] \\
&\quad\cdot\left[\prod_{i\in\omega^*\cup\Omega'}\left(g_2 T_i\right)^{r_i'}\right]\left(g^{J(m)} g_2^{F(m)}\right)^{\widehat{r}_s} \\
&= g_2^{a}\left[\prod_{i\in\omega'\cup\Omega'}\left(g_2 T_i\right)^{r_i\Delta_{i,S}(0)}\right]\left[\prod_{i\in\omega^*\cup\Omega'}\left(g_2 T_i\right)^{r_i'}\right] \\
&\quad\cdot\left(g^{J(m)} g_2^{F(m)}\right)^{r_s - a/F(m)} \\
&= g_2^{a}\left[\prod_{i\in\omega'\cup\Omega'}\left(g_2 T_i\right)^{r_i\Delta_{i,S}(0)}\right] \\
&\quad\cdot\left[\prod_{i\in\omega^*\cup\Omega'}\left(g_2 T_i\right)^{r_i'}\right]\left(g^{J(m)} g_2^{F(m)}\right)^{r_s} \\
&= g_2^{a}\left[\prod_{i\in\omega'\cup\Omega'}\left(g_2 T_i\right)^{r_i\Delta_{i,S}(0)}\right] \\
&\quad\cdot\left[\prod_{i\in\omega^*\cup\Omega'}\left(g_2 T_i\right)^{r_i'}\right]\left(u'\prod_{j\in\mathscr{M}} u_j^{m_j}\right)^{r_s} \\
&= \left[\prod_{i\in\omega'\cup\Omega'} d_{i0}^{\Delta_{i,S}(0)}\right]\left[\prod_{i\in\omega^*\cup\Omega'}\left(g_2 T_i\right)^{r_i'}\right]\left(u'\prod_{j\in\mathscr{M}} u_j^{m_j}\right)^{r_s}, \\
S_2 &= \left\{d_{i1}^{\Delta_{i,S}(0)} g^{r_i'}\right\}_{i\in\omega'\cup\Omega'}, \qquad S_2 = \left\{g^{r_i'}\right\}_{i\in\omega^*/\omega}, \\
S_3 &= g_1^{-1/F(m)} g^{r_s}.
\end{aligned} \tag{21}$$

Output. Finally, $\mathscr{A}$ outputs a signature $\sigma^* = (S_1^*, \{\sigma_{k_i}^*\}_{i\in\omega}, S_3^*)$ on some message M^* with public keys $A^{(1)}, \ldots, A^{(l)}$ for some l, where $A^{(1)}$ is equal to A as the challenge key. Attribute set $\omega_c^* \cup \Omega_c'$ contains the attribute from user 2 to l. Attribute set $\omega^* \cup \Omega'$ contains the attribute from user 1 to l. It outputs the private key for all keys except the challenge key. Algorithm $\mathscr{B}$ sets

$$\begin{gathered}
S_1 = S_1^*\left(\prod_{i\in\omega_c^*\cup\Omega_c'} e\left(g_2 T_i, p\sigma_{k_i}\right)\right)^{-1}\prod_{k=2}^{l} e\left(g,g\right)^{-\alpha^{(k)}}, \\
S_3 = S_3^*.
\end{gathered} \tag{22}$$

Then we have

$$\begin{aligned}
&\frac{e\left(g,S_1\right)}{e\left(u'\prod_{j\in\mathscr{M}} u_j^{m_j}, S_3\right)\prod_{i\in\omega_1^*\cup\Omega_1'} e\left(g_2 T_i, p\sigma_i\right)} \\
&= e\left(g,S_1\right)\left(\left[\prod_{i\in\omega_c^*\cup\Omega_c'} e\left(g_2 T_i, p\sigma_i\right)\right] e\left(u'\prod_{j\in\mathscr{M}} u_j^{m_j}, S_3\right)\right. \\
&\qquad\left.\cdot\prod_{i\in\omega_1^*\cup\Omega_1'} e\left(g_2 T_i, \sigma_i\right)\right)^{-1}\cdot\prod_{k=2}^{l} e\left(g,g\right)^{-\alpha^{(k)}} \\
&= \frac{e\left(g,S_1\right)\cdot\prod_{k=2}^{l} e\left(g,g\right)^{-\alpha^{(k)}}}{\left[\prod_{i\in\omega^*\cup\Omega'} e\left(g_2 T_i, p\sigma_k\right)\right] e\left(u'\prod_{j\in\mathscr{M}} u_j^{m_j}, S_3\right)} \\
&= \prod_{k=1}^{l} A^{(k)}\cdot\prod_{k=2}^{l} A^{-(k)} = A^{(1)} = A.
\end{aligned} \tag{23}$$

If the equation holds and $\omega = \omega'$ and $F(m^*) = 0(\text{mod } p)$, $\mathscr{B}$ computes and outputs:

$$\frac{S_1^*}{\prod_{i\in\omega_1\cup\Omega_1'}(S_2^*)^{f(i)}(S_3^*)^{J(m)}} = g^{ab}, \quad (24)$$

where

$$\begin{aligned} S_1^* &= g_2^a \prod_{i\in\omega_1^*\cup\Omega_1^*} (g_2 T_i)^{r_i\Delta_{i,S}(0)+r_i'} \left(u' \prod_{j\in\mathscr{M}} u_j^{m_j}\right)^{r_s} \\ &= g_2^a \prod_{i\in\omega_1^*\cup\Omega_1^*} \left(g^{f(i)r_i\Delta_{i,S}(0)} g^{f(i)r_i'}\right)\left(g^{J(m)}\right)^{r_s}, \\ S_2^* &= g^{r_i\Delta_{i,S}(0)+r_i'}, \qquad S_3^* = (g)^{r_s}. \end{aligned} \quad (25)$$

This is the solution to the given CDH problem.

We will analyze the probability of $\mathscr{B}$ without aborting to complete the description of the simulation. We require that the following cases happen.

We define the events A_k, A^*, B, C without abort during *Extract* queries, *Sign* queries,

$$\begin{aligned} A_k &: F(m_k) \neq 0 (\text{mod } l_m), \\ A^* &: F(m^*) = 0 (\text{mod } p). \end{aligned} \quad (26)$$

From the analysis above, the probability of $\mathscr{B}$ not aborting is

$$\Pr[\text{Not-abort}] \geq \Pr\left[\bigwedge_{k=1}^{q_I} A_k \wedge A^*\right]. \quad (27)$$

The assumption $l_m(n_m + 1) < p$ implies that if $F(m^*) = 0(\text{mod } p)$, then $F(m^*) = 0(\text{mod } l_m)$. Consider

$$\begin{aligned} &\Pr[A^*] \\ &= \Pr[F(m^*) = 0(\text{mod } p) \wedge F(m^*) = 0(\text{mod } l_m)] \\ &= \Pr[F(m^*) = 0(\text{mod } l_m)] \\ &\quad \cdot \Pr[F(m^*) = 0(\text{mod } p) \wedge F(m^*) = 0(\text{mod } l_m)] \\ &= \frac{1}{l_m(n_m+1)}. \end{aligned} \quad (28)$$

We also have that

$$\begin{aligned} \Pr\left[\bigwedge_{k=1}^{q_I} A_k \mid A^*\right] &= 1 - \Pr\left[\bigvee_{k=1}^{q_I} \overline{A}_k \mid A^*\right] \\ &\geq 1 - \sum_{k=1}^{q_I} \Pr\left[\overline{A}_k \mid A^*\right]. \end{aligned} \quad (29)$$

Since the output of $F(m_{i_1})$ and $F(m_{i_2})$ $(i_1 \neq i_2)$ will differ at least one random chosen value, the event $F(m_{i_1}) = 0(\text{mod } l_m)$ and $F(m_{i_2}) = 0(\text{mod } l_m)$ are independent. The event A_i and A^* are independent for any i. Hence, we have

$$\Pr\left[\bigwedge_{k=1}^{q_I} A_k \wedge A^*\right] \geq \frac{1}{l_m(n_m+1)}\left(1 - \frac{q_S}{l_m}\right). \quad (30)$$

Let $l_m = 2q_S$ and we get

$$\begin{aligned} \Pr[\text{Not-abort}] &\geq \Pr\left[\bigwedge_{k=1}^{q_I} A_k \wedge A^*\right] \\ &\geq \Pr\left[\bigwedge_{k=1}^{q_I} A_k \wedge A^*\right] \\ &\geq \frac{1}{4(n_m+1)q_S}. \end{aligned} \quad (31)$$

If the simulation does not abort, the probability for correct guess of $d-k$ elements subset Ω^* from $d-1$ element set Ω is $1/\binom{d-1}{d-k}$. Therefore, the advantage for solving CDH problem is

$$\epsilon' \geq \frac{\epsilon}{4\binom{d-1}{d-k}(n_m+1)(q_S)}. \quad (32)$$

Algorithm $\mathscr{B}$'s running time is of $\mathscr{A}$ plus the overhead in handling A's q_S *Sign* queries. The time complexity of $\mathscr{B}$ is

$$\begin{aligned} t' = t &+ \mathscr{O}(d(q_e + q_S) + n_m \cdot q_S)\rho \\ &+ (d(q_e + q_S) + n_m \cdot q_S)\tau \end{aligned} \quad (33)$$

where ρ and τ are the time for a multiplication and an exponentiation in $\mathbb{G}$, respectively. □

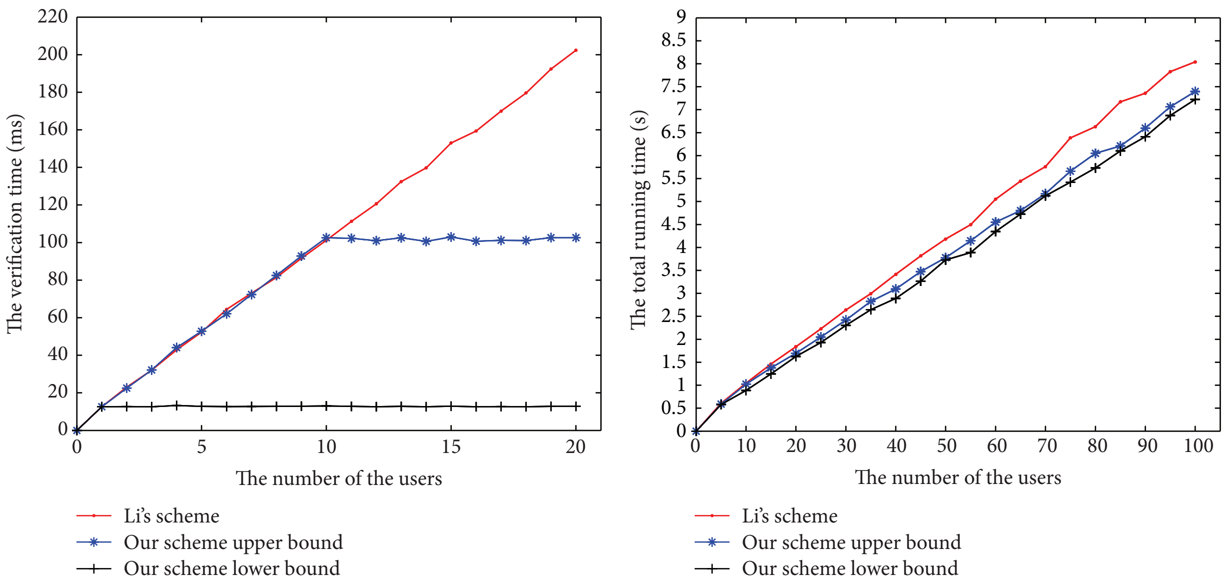
6. Performance Analysis

To analyze the performance of our proposed cryptosystem, we compare our ABMS scheme with Li et al.'s scheme in terms of storage, communication, and computational overheads. We define each type of overheads as follows.

Storage Overhead. The number of key materials holds by each entity and the size of signatures which are stored in the data center.

Computation Overhead. The computation resources which are occupied by the verifier and the total system.

6.1. Storage Overhead. Storage overheads are categorized into following types: the number of public parameters (*params*), private key available in the system, the number of private key $D^{(k)}$ $(k = 1 : l)$ which is held by each signing owner, and the size of signature storage in the data center. The total length of public parameters is smaller than Li's scheme. The length of private key held by each signer is the same as Li's scheme. The signature size stored in the data center is greatly decreased by using ABMS scheme than Li's scheme. The signature length of Li's scheme increases linear growth along with the number of users. While in our ABMS scheme, the lower bound of signature size is associated with the signer who have the maximum number of the attributes compared with other signers. The upper bound of the signature is associated with the number of universal attributes involved in the system. We



(a) The verification time

(b) The total running time

Figure 2: The simulation results.

Table 1: Performance analysis.

Functionality/ scheme	Signature size	Verification
Li et al. [10]	$2l+\sum_{i=1}^{l}\lvert c_i+d_i-u_i\rvert$	$\mathcal{O}\left(2l+\sum_{i=1}^{l}(c_i+d_i-u_i)\right)\varphi$
Our upper bound	$2+\lvert\mathcal{U}\rvert$	$\mathcal{O}(2+\lvert\mathcal{U}\rvert)\varphi$
Our lower bound	$2+\max_{1\le i\le l}\lvert c_i+d_i-u_i\rvert$	$\mathcal{O}\left(2+\max_{1\le i\le l}(c_i+d_i-u_i)\right)\varphi$

can aggregate l users signature into one short signature which can greatly decrease the storage overhead in the data center, especially when the number of uses involved in the system is huge. Here we compare our scheme with other schemes [23]. We let l be the number of signer, $|c_i + d_i - u_i|$ the size of the attribute set $\omega_i^* \cup \Omega_i^l$, and $|\mathcal{U}|$ the size of the universal of attribute set. φ is pairing running time in the *MultiVerify* algorithm. We make the comparison to list in Table 1. In the next section, we use a real workstation to simulate the ABMS scheme.

6.2. Computation Overhead. Li's scheme uses hash function to calculate the attribute. While in our ABMS scheme, we use T_i to construct ABMS scheme which can be proved in the standard model. The number of exponentiation to calculate T_i is associated with the security parameter. When two signers have the same attribute, *MComb* algorithm increases one more multiplication but decreases one pairing computation for the verifier by running *MultiVerify* algorithm. The computation cost of multiplication operation is greatly lower than the pairing operation. The computation cost for verification node to verify the signature can be greatly decreased because of the less pairing operations. The total computation cost of the whole system is also decreased because the multiplication operation cost is lower than the pairing operation.

6.3. The Performance Measurements. We now provide some information on the performance achieved by PBC (Pairing-Based Cryptography) library underlying pairing-based cryptosystems. In our experiment, the process is implemented on a workstation with an Inter Pentium CPU running at 2.40 GHz, 6 GB of RAM, and a 5400 RPM 320 GB Serial ATA drive. The OS on the test machine is Ubuntu 12.04 LTS 64-bits with kernel version 3.2.0-23-generic. We use type A pairings which are constructed on the 160-bits elliptic curve group based on the supersingular curve $y^2 = x^3 + x$ over a 512-bits finite field. On the test machine, we begin by estimating the cost in terms of basic cryptographic operations. The compute pairings in approximately 1.389 ms and exponentiations in $\mathbb{G}$ and $\mathbb{G}_T$ take about 1.994 ms and 0.187 ms, and multiplication in $\mathbb{G}$ and $\mathbb{G}_T$ takes about 0.005 ms and 0.002 ms. All of the computation is running by 10000 times for average. In our simulation system, there are 100 signers involved in the system and the total number of the attributes initialized by the *Setup* algorithm is 70. The maximum number of attributes belonging to individual signer is 7. We test the total running time and verification time between our ABMS scheme and Li's scheme [10] and we make the comparison in Figure 2. In Figure 2(a), we show the ABMS scheme's upper and lower bound of verification time and Li's ABS verification time. If all the users in the system share the same attribute set, the black line can be achieved which indicate the lower bound verification time of our ABMS scheme. If the attributes associated with users are all different, the blue line can be achieved which indicate the upper bound verification time of our ABMS scheme. When we run the *Mcomb* algorithm, it will introduce some multiplication in $\mathbb{G}$. Because the cost of multiplication in $\mathbb{G}$ is greatly smaller than the exponentiation and pairing operations, it will not bring much computation cost to the system. When total numbers of attributes belonging to different users do not reach the number of universe attribute set, the verification time of ABMS scheme is almost similar to the original Li's scheme. If the number of attribute belonging to a group of signers reached the number of universe attributes set, the verification time can be greatly decreased due to the less pairing computation. In Figure 2(b), we test the total running time between ABMS scheme and Li's scheme. The analysis is similar to the verification time analytic. From both verification and total running time, this algorithm can greatly decrease the computation cost as it can be showed in our simulation.

7. Conclusion

In this paper, we propose a scheme called attribute based multisignature in order to verify integrity of data efficiently. The ABMS scheme can compress multiple signatures into a single one in order to reduce the bandwidth needed to transmit signatures and the space needed to storage them. It can also provide signer's anonymity in which we use attributes set instead of identity. Our ABMS scheme is secure against existential unforgeability in an adaptive chosen-message attack under CDH problem. Even more important, ABMS scheme is more appropriate for wireless network communication than traditional ABS scheme.

Disclosure

A preliminary version of this paper has been presented in the 5th International Conference on Intelligent Networking and Collaborative Systems, INCoS 2013, pp. 173–180 [22].

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research is supported by Changjiang Scholars and Innovative Research Team in University (IRT1078); the Key Program of NSFC-Guangdong Union Foundation under Grant no. U1135002; Major National S&T Program (2011ZX03005002, 2012ZX03001009); the Fundamental Research Funds for the Central Universities (JY10000903001, K5051301017); the National Natural Science Foundation of China (61303218, 61370078, 61402109). The authors thank the sponsors for their support and the reviewers for helpful comments.

References

[1] T. Week, "Adobe says 2.9 m customers at risk after hacking," 2013, http://www.theweek.co.uk/technology/55448/adobe-says-29m-customers-risk-after-hacking.

[2] S. Shahandashti and R. Safavi-Naini, "Threshold attribute-based signatures and their application to anonymous credential systems," in *Progress in Cryptology—AFRICACRYPT 2009*, pp. 198–216, 2009.

[3] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology—EUROCRYPT 2005: Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005*, vol. 3494 of *Lecture Notes in Computer Science*, pp. 457–473, Springer, Berlin, Germany, 2005.

[4] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06)*, pp. 89–98, ACM, November 2006.

[5] X. A. Wang, X. Yang, M. Zhang, and Y. Yu, "Cryptanalysis of a fuzzy identity based encryption scheme in the standard model," *Informatica*, vol. 23, no. 2, pp. 299–314, 2012.

[6] P. Yang, Z. Cao, and X. Dong, "Fuzzy identity based signature," IACR Cryptology ePrint Archive Report 2008/002, 2008.

[7] S. F. Shahandashti and R. Safavi-Naini, "Threshold attribute-based signatures and their application to anonymous credential systems," Report 2009/126, ACR Cryptology ePrint Archive, 2009.

[8] H. K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-based signatures: achieving attribute-privacy and collusionresistance," in *IACR Cryptology ePrint Archive*, p. 328, 2008.

[9] J. Li and K. Kim, "Hidden attribute-based signatures without anonymity revocation," *Information Sciences*, vol. 180, no. 9, pp. 1681–1689, 2010.

[10] J. Li, M. H. Au, W. Susilo, D. Xie, and K. Ren, "Attribute-based signature and its applications," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communication Security (ASIACCS '10)*, pp. 60–69, ACM, April 2010.

[11] D. Cao, B. Zhao, X. Wang, and J. Su, "Flexible multi-authority attribute-based signature schemes for expressive policy," *Mobile Information Systems*, vol. 8, no. 3, pp. 255–274, 2012.

[12] K. Itakura and K. Nakamura, "A public-key cryptosystem suitable for digital multisignatures," *NEC Research and Development*, vol. 71, pp. 1–8, 1983.

[13] T. Okamoto, "A digital multisignature scheme using bijective public-key cryptosystems," *ACM Transactions on Computer Systems*, vol. 6, no. 4, pp. 432–441, 1988.

[14] K. Ohta and T. Okamoto, "A digital multisignature scheme based on the Fiat-Shamir scheme," in *Advances in Cryptology—ASIACRYPT '91*, vol. 739 of *Lecture Notes in Computer Science*, pp. 139–148, Springer, Berlin, Germany, 1993.

[15] T. Okamoto, "Multi-signature schemes secure against active insider attacks," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 82, no. 1, pp. 21–31, 1999.

[16] S. Micali, K. Ohta, and L. Reyzin, "Accountable-subgroup multisignatures," in *Proceedings of the 8th ACM Conference on Computer and Communications Security (CCS '01)*, pp. 245–254, Philadelphia, Pa, USA, November 2001.

[17] A. Boldyreva, "Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme," in *Public Key Cryptography—PKC 2003: Proceedings of the 6th International Workshop on Practice and Theory in Public Key Cryptography Miami, FL, USA, January 6–8, 2003*, vol. 2567 of *Lecture Notes in Computer Science*, pp. 31–46, Springer, Berlin, Germany, 2002.

[18] S. Lu, R. Ostrovsky, A. Sahai, H. Shacham, and B. Waters, "Sequential aggregate signatures and multisignatures without random oracles," in *Advances in Cryptology-EUROCRYPT*, pp. 465–485, 2006.

[19] J. H. Cheon, Y. Kim, and H. Yoon, "A new id-based signature with batch verification," in *IACR Cryptology ePrint Archive*, p. 131, 2004.

[20] C. Gentry and Z. Ramzan, "Identity-based aggregate signatures," in *Public Key Cryptography*, vol. 3958 of *Lecture Notes in Computer Science*, pp. 257–273, Springer, Berlin, Germany, 2006.

[21] X. Liu, J. Ma, Q. Li, J. Xiong, and F. Huang, "Attribute based multi-signature scheme in the standard model," in *Proceedings of the 9th International Conference on Computational Intelligence and Security (CIS '13)*, pp. 738–742, IEEE, December 2013.

[22] X. Liu, T. Zhang, J. Ma, H. Zhu, and F. Cai, "Efficient data integrity verification using attribute based multi-signature scheme in wireless network," in *Proceedings of the 5th IEEE International Conference on Intelligent Networking and Collaborative Systems (INCoS '13)*, pp. 173–180, IEEE, September 2013.

[23] J. Li, Q. Wang, C. Wang, and K. Ren, "Enhancing attribute-based encryption with attribute hierarchy," *Mobile Networks and Applications*, vol. 16, no. 5, pp. 553–561, 2011.