Recommended Paper

A CDH-based Ordered Multisignature Scheme Provably Secure without Random Oracles

Naoto Yanai[1,a)] Eikoh Chida[2,b)] Masahiro Mambo[3,c)] Eiji Okamoto[1,d)]

Received: May 27, 2013, Accepted: October 9, 2013

Abstract: Ordered multisignature scheme is a signature scheme to guarantee both validity of an electronic document and its signing order. Although the security of most of such schemes has been proven in the random oracle model, the difficulty of implementation of the random oracle implies that the security should be proven without random oracles, i.e., in the standard model. A straightforward way to construct such schemes in the standard model is to apply aggregate signature schemes. However, the existing schemes based on the CDH problem are inefficient in the sense that the number of computations of the bilinear maps and the length of public keys depend upon the length of (a hash value of) the message. Therefore, in this paper, we propose a CDH-based ordered multisignature scheme which is provably secure in the standard model under a moderate attack model. Its computational cost for the bilinear maps and the size of public key are independent of the length of (a hash value of) the message. More specifically, in comparison with the existing schemes, the public key length is reduced to three group elements from 512 group elements while the computational cost is reduced to 0.85 msec from 1.6 msec.

Keywords: multisignatures, ordred multisignatures, standard model, CDH problem

1. Introduction

1.1 Background

Multisignature scheme [15] is a digital signature scheme that compresses n given signatures on a common message from n signers into a single signature, and it is a primitive suitable for small devices with low computational power or a small amount of storage. An *ordered multisignature* [11] scheme is one in which the signing order of the signers is recorded and both the validity of the message and the signing order are verified. This primitive is useful for various applications such as interdomain routing protocols. As described in Section 7, one of possible applications is for securing border-gateway protocol (BGP) [24], named *data-plane security* [12]. One of requirements for BGP is a policy [12] where operators specify which routes are advertised to neighboring networks, and the data-plane security is aimed to reject packets through invalid routes that are not advertised. Intuitively, the packets should be sent via only routers matching the policy, and the order verifiability of ordered multisignatures is useful for verifying actual packet forwarding. Although the security of the most ordered multisignature schemes has been proven in the random oracle model [3], Canetti et al. showed a negative result [10] that there exist signature and encryption schemes, that are secure in the random oracle model but for which any implementation of the random oracle results in insecure schemes. This result implies that the security should be proven without random oracles, and such a model is called *standard model*.

According to Ref. [6] one may imagine that an ordered multisignature scheme can be simply constructed from aggregate signatures [8], which are digital signatures where any party given n signatures of n messages from n users can combine all of these signatures into a single short signature. Particularly, each document is concatenated with the signer's position in the group and then signed using an aggregate signature scheme. These generated signatures are then aggregated into an ordered multisignature. Several aggregate signature schemes have been proposed in the standard model, and ordered multisignature schemes are constructed from these schemes. In this approach, Ahn et al. [1] and Lu et al. [18] have respectively proposed aggregate signature schemes based on CDH problem. However, these schemes are impractical. In the scheme by Ahn et al. the number of the computationally heavy bilinear map operations increases linearly with the length of (a hash value of) the message. In the scheme by Lu et al. the size of the public key is larger than the packet limitation of BGP.

1.2 Our Contributions

In this paper, we propose a secure ordered multisignature scheme in the standard model, that is not constructed from aggregate signature schemes, under a moderate attack model as long as the CDH assumption holds. Our proposed scheme is efficient

[1] University of Tsukuba, Tsukuba, Ibaraki 305–8573, Japan
[2] Ichinoseki National College of Technology, Ichinoseki, Iwate 021–8511, Japan
[3] Kanazawa University, Kanazawa, Ishikawa 920–1192, Japan
[a)] yanai@cipher.risk.tsukuba.ac.jp
[b)] chida@ichinoseki.ac.jp
[c)] mambo@ec.t.kanazawa-u.ac.jp
[d)] okamoto@risk.tsukuba.ac.jp

The initial version of this paper was presented at Computer Security Symposium 2012 (CSS 2012) in October 2012. This paper was recommended to be submitted to Journal of Information Processing (JIP) by Program Chair of CSS 2012.

in terms of both the communication and the computational costs, and we believe that the proposed scheme is the most practical one for possible applications described in Section 7.

For the communication cost, we note that the signature size is independent of the number of signers, and the signature always consists of three elements of a group $\mathbb{G}$. The length of the public key is independent of the length of (a hashed value of) the message and also consists of three elements in $\mathbb{G}$. Most CDH-based schemes are based on the Waters signature scheme [29], and the public keys in those schemes depend on the length of the message. The length of the message is based on the binary length of the elements in $\mathbb{G}$, and there are 160 elements in each public key with 80-bit security and 512 elements for 128-bit security. Hence, the communication efficiency is diminished in trivial constructions based on the Waters signature scheme. In contrast, in our scheme, by removing the Waters hash function from the public keys, we reduced the number of elements for the public key to less than 1%: to three elements (1,536 bits) from 512 elements (263,168 bits).

For the computational cost, we estimate the total computations cost, including both signing and verification. There is a high computational cost for the bilinear maps used in verification, but in our scheme, the number of computations of the bilinear maps is independent of both the message length and the number of signers: our scheme requires only four bilinear map computations, regardless of the length of the message or the number of signers. The time for each computation of bilinear maps is about 6.398 msec, even with the latest software library and hardware, such as the University of Tsukuba Elliptic Curve and Pairing Library (TEPLA) [14] and a Core i7 3960X. As described above, multisignatures are primarily of use for devices with low computational power and little storage, and so it is important to reduce the number of operations with a high computational cost. The scheme by Ahn et al. [1] requires eleven bilinear map computations, but our scheme requires only four bilinear map computations. We therefore estimate the total computational cost for our scheme will be about half of that for the existing scheme which is available for a realistic scenario.

Our construction is based on the following observation. Conditions to construct an ordered multisignature scheme secure in the standard model under the CDH assumption is to be a signature scheme such that a signature consists of three separable components, i.e., a three-partitioned signature. The three-partitioned signature allows us to remove the Waters hash from the public key and also to use the technique of the existing efficient scheme. The security can be proven by the technique of the existing schemes which is utilized as building blocks in our construction.

As a possible application of the proposed scheme, we describe the following scenario. The application is for securing the BGP; in this application, signers correspond to routers. According to Sriram et al. [27], routers have little computational power, which makes it difficult for them to sign and to verify signatures. In this application, since our scheme has a smaller public key than the packet limitation of the BGP in comparison with the existing scheme whose public key size is larger than the packet limitation of BGP. In addition, the number of bilinear map computations is independent of the number of signers and the message length, we believe that the efficiency of our proposed scheme makes the application possible. We introduce the detail of the application in Section 7.

This is a full version of Ref. [30] we published in the Computer Security Symposium (CSS) 2012.

1.3 Related Work

The multisignature scheme was proposed by Itakura and Nakamura [15], and the security was formalized by Ohta and Okamoto [23] and Boldyreva [5]. The aggregate signature scheme, which is a generalized multisignature, was proposed by Mitomi and Miyaji [20], and Boneh et al. [8] proposed a scheme with bilinear maps. Ordered multisignature schemes can be constructed from aggregate signature schemes, and schemes such as RSA, which are based on permutations, seem to be more efficient and thus have a lower computational cost. However, the permutation-based schemes [2], [9], [19], [22] do not support the order flexibility defined by Mitomi and Miyaji [20], and so the number of signers is impossibly restricted. In contrast, the existing schemes based on bilinear maps support the order flexibility and the same public key can be used for all signing order.

Several schemes in the standard model, including aggregate signature schemes, have been proposed [1], [17], [18], [25], [26]. To increase security, the scheme should be based on the well-studied problem, such as the CDH problem, which are found only in Refs. [1], [18]. However, the number of computations of the bilinear maps in the scheme by Ahn et al. [1] and the number of elements of the public key in the scheme by Lu et al. [18] both increase linearly with the length of the message.

As an investigation of security of the S-BGP, Boldyreva and Lychev [7] formalized it and proved the relation between the unforgeability of the signatures and the security.

2. Preliminaries

In this section, we present some background on groups with bilinear maps and its security assumption.

2.1 Notations

Let the number of signers be n. We denote by V a verifier, by m a message to be signed, by m_i the i-th bit of the message m, by σ_i the signature generated by the i-th signer, by pk_i the public key of the i-th signer, by sk_i the secret key of the i-th signer and by $a \parallel b$ a concatenation of elements a and b, where the concatenation can be easily divided into the original elements a and b. We define $\psi_i := pk_1 \parallel \cdots \parallel pk_i$ to be the signing order from the first signer to the i-th signer, and denote by $|\psi_i|$ the number of signers in ψ_i.

2.2 Bilinear Maps

Let $\mathbb{G}$ and $\mathbb{G}_T$ be groups with the same prime order p. We then define bilinear maps and bilinear groups as follows:

Definition 1 (Bilinear maps)**.** A bilinear map $\boldsymbol{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is a map such that the following conditions hold, where g is a generator of $\mathbb{G}$: (Bilinearity) For all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p^*$, $\boldsymbol{e}(u^a, v^b) = \boldsymbol{e}(u, v)^{ab}$; (Non-degeneracy) For any generator $g \in \mathbb{G}$, $\boldsymbol{e}(g, g) \neq 1_{\mathbb{G}_T}$, $1_{\mathbb{G}_T}$ is an identity element over $\mathbb{G}_T$; (Computable)

There is an efficient algorithm to compute $\boldsymbol{e}(u, v)$ for any $u, v \in \mathbb{G}$.

In this paper, we say that $\mathbb{G}$ is a bilinear group if all these conditions hold, and we assume that the discrete logarithm problem (DLP) in bilinear groups is hard. We call the parameter $(p, \mathbb{G}, \mathbb{G}_T, \boldsymbol{e})$ *pairing parameter*.

2.3 Security Assumption

The CDH assumption is defined as follows.

Definition 2 ((t, ϵ)-CDH assumption in $\mathbb{G}$)**.** We define the CDH problem in bilinear groups with a security parameter 1^k as follows: for a given $(g, g^a, g^b) \in \mathbb{G}$ with uniformly random $a, b \in \mathbb{Z}_p$ and a pairing parameter $(p, \mathbb{G}, \mathbb{G}_T, \boldsymbol{e})$ as input, compute g^{ab}, where p is a prime order of $\mathbb{G}$. We say the (t, ϵ)-CDH assumption holds in $\mathbb{G}$ if and only if there is no probabilistic polynomial-time algorithm that can solve the CDH problem in $\mathbb{G}$ with the execution time t with probability greater than ϵ.

2.4 Waters Hash Function

The Waters hash function [29] is the following function consisting $\ell + 1$ generators, where ℓ is the length of a message. A scheme based on the CDH problem in the standard model can be constructed via the security proof utilizing this function.

Waters Hash Function: Choose $\ell + 1$ generators $(u', u'_1, \cdots, u'_\ell) \in \mathbb{G}^{\ell+1}$. Then, for all message $m = (m_1, \cdots, m_l) \in \{0, 1\}^\ell$, compute $H(m) = u' \prod_{i=1}^{\ell} u_i^{m_i}$.

The collision resistance was proven in Ref. [13]. In the scheme by Lu et al. [18], each signer has the personalized one, which is with trapdoors $(x', x_1, \cdots, x_\ell)$ such that $u_i = g^{x_i}$ holds for any generator $g \in \mathbb{G}$, as a part of its own public key. However, such a construction increases the length of the public key in proportion to the message length.

2.5 Signature Scheme by Waters

In this section, we recall the signature scheme by Waters [18], which is a variant of the Waters signature scheme [29]. The scheme utilizes the Waters hash function described above, and hence a message in the scheme will be dealt as a bit-string of the form $\{0, 1\}^\ell$ for all ℓ.

Setup: Given a security parameter 1^k, generate a pairing parameter $(p, \mathbb{G}, \mathbb{G}_T, \boldsymbol{e})$, and choose a random generator $g_1, g_2 \in \mathbb{G}$ and $\ell + 1$ generators $u', u_1, \cdots, u_\ell \in \mathbb{G}$. Then, choose a random number $\alpha_W \leftarrow \mathbb{Z}_p$, and compute $A_W \leftarrow \boldsymbol{e}(g, g)^{\alpha_W}$. Output $(p, \mathbb{G}, \mathbb{G}_T, \boldsymbol{e}, g_1, g_2, u', u_1, \cdots, u_\ell, A_W)$ as a public key pk and $g_2^{\alpha_W}$ as a corresponding secret key.

Signing: Given sk and a message m, parse m as a bit-string $(m_1, \cdots, m_\ell) \in \{0, 1\}^\ell$. Pick a random number $r \leftarrow \mathbb{Z}_p$ and compute as follows:

$$\begin{aligned} S &\leftarrow g_2^{\alpha_W} \left(u' \prod_{j=1}^{\ell} u_j^{m_j} \right)^r, \\ R &\leftarrow g_1^r. \end{aligned}$$

Output $\sigma = (S, R)$ as a signature on m.

Verification: Given m, σ and pk, parse m as a bit-string $(m_1, \cdots, m_\ell) \in \{0, 1\}^\ell$ and σ as (S, R). Check that the following equation holds:

$$\boldsymbol{e}(S, g_1) \cdot \boldsymbol{e}\left(R, \left(u' \prod_{j=1}^{\ell} u_j^{m_j}\right)\right)^{-1} \stackrel{?}{=} A_W.$$

If the equation holds, output *accept*. Otherwise, output *reject*.

In Ref. [29], Waters has proven that this scheme is secure if the CDH assumption holds.

3. Ordered Multisignature

In this section, we explain a syntax of the ordered multisignature schemes, define its security, and describe a technical problem for constructing the ordered multisignatures. Hereafter, we assume that each user has a single public key, and denote by pk_i i-th signer without loss of generality.

3.1 The Syntax

Ordered multisignature scheme consists of the following algorithms, where i satisfies $1 \leq i \leq n$.

Setup: Given security parameter 1^k, generate a public parameter *para*.

Key Generation: Given *para*, generate a secret key sk_i and its corresponding public key pk_i.

Signing: Given a secret key sk_i, a public key pk_i, a message m, a multisignature σ' from the previous signers, and a signing order ψ_{i-1}, generate a signature σ. Finally, set $\psi_i = \psi_{i-1} \parallel pk_i$ and output the signature σ on m in ψ_i.

Verification: Given m, σ and ψ_n, output *accept* or *reject*.

Correctness: In an ordered multisignature scheme, we say that the scheme is correct if, for all *para*, sk_i and pk_i given by **Setup** and **Key Generation**, **Verification**(m, **Signing**($sk_i, pk_i, m, \sigma', \psi_{i-1}$), ψ_i) outputs *accept*.

3.2 Security Model

In this model, there exists an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. Our model is a variant of the certified key model [5]. The certified key model is a model supported by public key infrastructure (PKI) and assumes that each user knows a secret key corresponding its own public key [*1]. More precisely, $\mathcal{C}$ has a list $\mathcal{L}$ of certified keys that is used to register users and their own public keys. $\mathcal{A}$ can know all secret keys corresponding to public keys included in $\mathcal{L}$ except for the one given by $\mathcal{C}$. $\mathcal{A}$'s advantage is equal to the probability that $\mathcal{C}$ outputs *accept* in the following game. We will denote by $x^{(i)}$ the value of the i-th query for all x. Similarly as the existing model of ordered multisignatures [6], our security model guarantees authenticity of the message signed by an honest signer and its position i in a path, but not which signers signed before or will sign after the i-th signer. We do not consider *switching* of the positions among colluding malicious signers. For instance as shown in Ref. [6], there are malicious signers corresponding to pk_1 and pk_3 colluding each other against an honest signer corresponding to pk_2. Signers corresponding to pk_1 and pk_3 may be able to compute some signature σ on m in $\psi_2 = pk_1 \parallel pk_2$ after obtaining σ^* on m in $\psi_3^* = (pk_3 \parallel pk_2) \parallel pk_1$. To the best of our

[*1] There exist models providing stronger security such that there is no such an assumption. However, PKI-based security based on the key registration model is realistic and constructing schemes secure under the stronger security is an open problem.

knowledge, there is no DLP-based scheme preventing the switching of the positions among colluding malicious signers. Namely, constructing such a DLP-based scheme remains an open problem. We note that, according to Ref. [6], this security model seems to be acceptable in the application described in Section 7.

Initial Phase: The challenger $\mathcal{C}$ generates a public parameter $para$ by **Setup** and a pair of challenger keys (sk^*, pk^*) by using **Key Generation**. Then, $\mathcal{C}$ initializes $\mathcal{L} := pk^*$, and runs $\mathcal{A}$ with $para$ and pk^* as input.

Certification Query: $\mathcal{A}$ generates (sk_i, pk_i), and sends them to $\mathcal{C}$. Then, $\mathcal{C}$ registers pk_i in $\mathcal{L}$ if sk_i is a secret key corresponding to pk_i.

Signing Query: For any i $(1 \leq i \leq n)$, $\mathcal{A}$ generates a signing query $(m^{(h)}, \sigma', \psi_{i-1}^{(h)})$ as the h-th query for the challenge key pk^*, where the following conditions hold for the query i satisfies $1 \leq i \leq n$: **Verification** algorithm outputs $accept$; $\psi_{i-1}^{(h)}$ does not include pk^*; for $j \in [1, i-1]$ pk_j in $\psi_{i-1}^{(h)}$ is included in $\mathcal{L}$; no pk_j appears more than once in $\psi_{i-1}^{(h)}$; $|\psi_{i-1}^{(h)}| < n$. The sining query is given by $\mathcal{A}$ to $\mathcal{C}$ together with the public key pk^*. $\mathcal{C}$ runs **Signing** $(sk^*, pk^*, m^{(h)}, \sigma', \psi_{i-1}^{(h)})$, and obtains σ and $\psi_i^{(h)} = \psi_{i-1}^{(h)} \parallel pk^*$. Finally, $\mathcal{C}$ returns σ on $m^{(h)}$ in $\psi_i^{(h)}$.

Output: After iterating over the above steps, $\mathcal{A}$ outputs a forgery $(m^*, \sigma^*, \psi_n^*)$. Here, let the target signer be the i^*-th signer in ψ_n^*, and let the following conditions hold for the forgery: **Verification**$(m^*, \sigma^*, \psi_n^*, \{pk_i\}_{i=1}^n)$ outputs $accept$; $m^* \notin \{m^{(h)}\}_{h=1}^{q_s} \vee \psi_{i^*-1}^* \notin \{\psi_{i-1}^{(h)}\}_{h=1}^{q_s}$ holds [*2], where $\psi_{i^*-1}^*$ is extracted from ψ_n^* as a signer structure from the first signer to the signer previous to the target signer; ψ_n^* includes pk^*; for $j \in [1, n]$, pk_j in ψ_n^* is included in $\mathcal{L}$; no pk_j appears more than once in ψ_n^*. If all these conditions hold, then $\mathcal{C}$ outputs $accept$. Otherwise, $\mathcal{C}$ outputs $reject$.

Definition 3. We say that an adversary $\mathcal{A}$ breaks an ordered multisignature scheme with $(t, q_c, q_s, \ell, n, \epsilon)$ if and only if a challenger $\mathcal{C}$ outputs $accept$, in the security game described above within the execution time t and with probability greater than ϵ. Here, $\mathcal{A}$ can generate at most q_c certification queries and at most q_s signing queries, ℓ is the length of the message output by $\mathcal{A}$, and n is the number of signers included in the forgery.

3.3 Difficulty of Constructing Ordered Multisignatures

Ordered multisignatures guarantee the signing order in addition to the validity of a message for standard multisignatures. In a trivial construction, this capability can be constructed from signing both the message and the signing order by utilizing the full-domain hash schemes such as the aggregate signature scheme by Boneh et al. [8]. However, the number of computations of the bilinear maps in such an approach becomes linear with respect to the number of signers. Although a scheme [18] using the Waters hashes with trapdoors as public keys is efficient in terms of the computational cost, the length of the public keys in the scheme becomes linear.

4. Proposed Scheme

4.1 Our Approach

In our approach, signature components are partitioned into three parts, i.e., each of which is related to either secret key, message or the signing order, and each part can be operated individually. We call such a construction "three-partitioned signature". This is for the following reasons: (1) taking the Waters hash into the public parameter from the public keys and (2) combining the Boldyreva et al.'s technique described below. The three-partitioned construction allows us to utilize both techniques by utilizing individual components.

We describe more details below. The reason why the length of the public key is long in the scheme by Lu et al. [18] is the use of the Waters hash functions with trapdoors described in Section 2.4. In order to reduce the length of the public keys, the Waters hash functions with trapdoors should be removed from the public keys. In this approach, we found a fact that the trapdoors of the Waters hash functions are unnecessary for compressing the signatures, because the messages is common strings among the signers. Namely, the Waters hash can be taken into the public parameter using only the message component from signers' public keys, and hence the public key length becomes independent of the message length. This construction also gives a benefit that the number of the computations of bilinear maps becomes independent of both the length of the message in comparison to the scheme by Ahn et al. [1].

Meanwhile, we need a technique to verify the signing order. The idea to address this is the second reason. Actually, we can use the technique of the ordered multisignature scheme proposed by Boldyreva el al. [6] in order to construct a scheme in the standard model, while the scheme by Boldyreva et al. is a construction in the random oracle model. Their scheme utilizes indexes representing the positions of signers to verify the signing order, and is efficient in the sense of the fixed number of computations of the bilinear maps. Here, we found that the random oracles are no longer required if we consider only a component of the signing order. Therefore, the number of the bilinear map computation is independent of both the message length and the number of the signers, while the capability for verifying the signing order is provided.

4.2 Construction of the Scheme

We assume that there exists a trusted center to generate the public parameter. A message m in this scheme will be dealt with as a bit-string of the form $\{0, 1\}^\ell$ for all ℓ. We can also let the message length ℓ be the output length of a collision-resistant hash function $H : \{0, 1\}^* \rightarrow \{0, 1\}^\ell$.

Setup: Given 1^k, this algorithm generates the pairing parameter $(p, \mathbb{G}, \mathbb{G}_T, \boldsymbol{e})$ described in Section 2.2 and generates random generators $g_1, g_2 \in \mathbb{G}$ and $\ell + 1$ generators $(u', u_1, \cdots, u_\ell) \in \mathbb{G}^{\ell+1}$. The algorithm outputs $(p, \mathbb{G}, \mathbb{G}_T, \boldsymbol{e}, g_1, g_2, u', u_1, \cdots, u_\ell)$ as the public parameter.

[*2] We should discuss $(m^*, \psi_{i^*-1}^*) \notin \{(m^{(h)}, \psi_{i-1}^{(h)})\}_{h=1}^{q_s}$ as a natural security requirement. However, until the publication of Ref. [30], there is no scheme achieving the requirement in the standard model, and constructing such a scheme is an open problem. In this paper, we discuss $m^* \notin \{m^{(h)}\}_{h=1}^{q_s} \vee \psi_{i^*-1}^* \notin \{\psi_{i-1}^{(h)}\}_{h=1}^{q_s}$ as in Ref. [30]. Even such a moderate model is not discussed in the standard model signature schemes [1], [18]. Through a discussion under this model, we prove that a proposed scheme guarantees the validity of messages signed by an honest signer and his/her positions in the signing order.

Key Generation: Given $(p, \mathbb{G}, \mathbb{G}_T, \boldsymbol{e}, g_1, g_2, u', u_1, \cdots, u_\ell)$, this algorithm chooses random numbers $\alpha_i, t_i, v_i \leftarrow \mathbb{Z}_p^*$, and sets $A_i = g_1^{\alpha_i}, T_i = g_1^{t_i}$ and $V_i = g_1^{v_i}$. The algorithm outputs $(g_2^{\alpha_i}, t_i, v_i)$ as a secret key sk_i and (A_i, T_i, V_i) as the corresponding public key pk_i.

Signing: Given $(sk_i, pk_i, m, \sigma', \psi_{i-1})$, this algorithm parses m as ℓ-bit strings $(m_1, \cdots, m_\ell) \in \{0,1\}^\ell$, the signature σ' as (S_{i-1}, R_{i-1}) and ψ_{i-1} as a set $\{pk_j\}_{j=1}^{i-1}$ of public keys, where $pk_j = (A_j, T_j, V_j)$ for all j. If $i = 1$, i.e., for a first signer in the signing group, then $(S_{i-1}, R_{i-1}) = (1, 1)$ and $\{pk_j\}_{j=1}^{i-1} = \emptyset$ are set [*3], and the following verification step is skipped. Next, the algorithm verifies that the received signature σ' is a valid signature on m in ψ_{i-1} by using **Verification** for $n = i - 1$. If **Verification** outputs *reject*, this algorithm aborts the process. Otherwise, the algorithm generates a random number $r_i \leftarrow \mathbb{Z}_p^*$ and computes the following:

$$S_i = S_{i-1} \cdot g_2^{\alpha_i} \left(u' \prod_{j=1}^{\ell} u_j^{m_j} \right)^{r_i} R_i^{it_i + v_i} \left(\prod_{pk_k \in \{pk_j\}_{j=1}^{i-1}} T_k^k V_k \right)^{r_i},$$

$$R_i = R_{i-1} \cdot g_1^{r_i}.$$

Finally, the algorithm sets $\psi_i = \psi_{i-1} \parallel pk_i$, then outputs m, $\sigma = (S_i, R_i)$.

Verification: Given $(m, \sigma, \psi_n, \{pk_i\}_{i=1}^n)$, this algorithm parses m as an ℓ-bit string $(m_1, \cdots, m_\ell) \in \{0,1\}^\ell$ and σ as (S_n, R_n). The algorithm extracts each signer's public key (A_i, T_i, V_i) from $\{pk_i\}_{i=1}^n$ and verifies that the following equation holds:

$$\boldsymbol{e}(S_n, g_1) \stackrel{?}{=} \boldsymbol{e}\left(g_2, \prod_{i=1}^{n} A_i\right) \boldsymbol{e}\left(R_n, u' \prod_{j=1}^{\ell} u_j^{m_j}\right) \boldsymbol{e}\left(R_n, \prod_{i=1}^{n} T_i^i V_i\right).$$

If not, the algorithm outputs *reject*. Otherwise, the output is *accept*.

4.3 More Construction

In some applications, even the size of the public parameter becomes a bottleneck. In order to reduce the parameter size, we can adopt the Naccache approach [21]. In this construction, the messages are divided into chunks of λ bits. The size of λ is 32 bits, and, for 128-bit security, the public parameter can be reduced from 256 generators to eight generators.

We do not provide the details of the scheme construction and the security proof, but these can be obtained similarly as Ref. [21]. The Naccache approach decreases the reduction cost in the security proof. Hence, we have to utilize the approach carefully.

5. Security Analysis

We now prove that the proposed scheme is secure in the standard model. The proof is similar as the Theorem 1 in Ref. [18] and the Theorem 3.3 in Ref. [6].

Theorem 4. The proposed ordered multisignature scheme is the $(t, q_c, q_s, \ell, n, \epsilon)$-secure if $(t_{CDH}, \epsilon_{CDH})$-CDH assumption in $\mathbb{G}$ holds, where

$$t_{CDH} = t + (3q_c + 9q_s + 2)\, t_e + 2q_c t_p, \qquad (1)$$

$$\epsilon_{CDH} = \frac{\epsilon}{16(\ell + 1) + e(q_s - 1)}, \qquad (2)$$

where t_e is the computational cost for one exponentiation computation, t_p is the computational cost for one bilinear map and e is the base of natural logarithm.

Proof. We describe a proof sketch. We construct an algorithm $\mathcal{B}$, given a challenge of the CDH problem, to solve the CDH problem. We assume that an adversary $\mathcal{A}$ who breaks the proposed scheme with $(t, q_c, q_s, l, n, \epsilon)$ exists. From the definition of the forgery, without the loss of generality, the output by $\mathcal{A}$ can be classified as follows:

(**case 1**): $m^* \notin \{m^{(h)}\}_{h=1}^{q_s}$;

(**case 2**): $m^* \in \{m^{(h)}\}_{h=1}^{q_s} \wedge \psi_i^* \notin \{\psi_i^{(h)}\}_{h=1}^{q_s}$.

For case 1, $\mathcal{B}$ generates a challenge in the Waters signature scheme [29], the Waters challenge for short, by using the challenge in the CDH problem, and then generates a challenge in the proposed scheme from the Waters challenge. On the other hand, for case 2, $\mathcal{B}$ directly generates a challenge in the proposed scheme from the CDH challenge without generating the Waters challenge. Then $\mathcal{B}$ runs $\mathcal{A}$ with the challenge in either case. We also analyze the probabilities and the execution time that $\mathcal{B}$ successes to solve the problem, (t', ϵ') for case 1 and (t'', ϵ'') for case 2. Then, we compute the whole probability ϵ_{CDH} and the whole computational time t_{CDH}.

Without the loss of generality, we assume that there exists exactly one signer, the target signer, for which $\mathcal{A}$ does not know the secret key. $\mathcal{B}$ has the list $\mathcal{L}$ of certified-keys, and for all x, we denote the value of the j-th query by $x^{(j)}$. $\mathcal{B}$ interacts with $\mathcal{A}$ as follows:

(**case 1**): We construct an algorithm $\mathcal{B}$ to break the Waters signature scheme using $\mathcal{A}$. This step is almost same as the proof in Ref. [18].

Lemma 5. The proposed scheme is $(t, q_c, q_s, l, n, \epsilon)$-secure if the Waters signature scheme is $(t_W, q_{W_s}, \epsilon_W)$ where q_{W_s} is the number of queries to the Waters signature scheme and $q_s = q_{W_s}$, $\epsilon_W = \epsilon$, and $t_W = t + (2q_c + 2q_s + 2)\, t_e + 2q_c t_p$.

Proof. $\mathcal{B}$ can access an oracle for the Waters signature scheme, $\mathcal{O}_W$, and interacts with $\mathcal{A}$ for case 1 as follows:

Initial Phase: Given a public parameter $(g_1, g_2, u', u_1, \cdots, u_\ell, p, \mathbb{G}, \mathbb{G}_T, \boldsymbol{e})$ and a challenge key A_W as a challenge of the Waters signature, $\mathcal{B}$ generates random numbers $(t^*, v^*) \leftarrow \mathbb{Z}_p$ and then sets $T^* = g_1^{t^*}$, $V^* = g_1^{v^*}$, $A^* = A_W$ as the public key pk^* of the target signer. Here, let its corresponding secret key be $g_2^{\alpha_W}$. Then, $\mathcal{B}$ runs $\mathcal{A}$ with $(p, \mathbb{G}, \mathbb{G}_T, \boldsymbol{e}, g_1, g_2, u', u_1, \cdots, u_\ell, A^*, T^*, V^*)$.

Certification Query: For any signer, $\mathcal{A}$ generates a secret key $sk_i = (g_2^{\alpha_i}, t_i, v_i)$ and its corresponding public key $pk_i = (A_i, T_i, V_i)$, and then provides sk_i and pk_i to $\mathcal{B}$. $\mathcal{B}$ checks that $\boldsymbol{e}(g_2^{\alpha_i}, g_1) = \boldsymbol{e}(g_2, A_i)$, $V_i = g_1^{v_i}$, and $T_i = g_1^{t_i}$. If all these equations hold, pk_i is registered to the public key list $\mathcal{L}$. Otherwise, the output is $\perp$.

Signing Query: This step is almost the same as that in Ref. [18]. After obtaining a signature (S', R') from $\mathcal{O}_W$, $\mathcal{B}$ sets

[*3] We assume that $\prod_{pk_k \in \{pk_j\}_{j=1}^{i-1}} T_k^k V_k = 1$ holds for $i = 1$.

$R_i = R'$ and computes as follows:

$$S_i = S' \times (R')^{\sum_{j=1}^{i} jt_j+v_j} \cdot g_2^{\sum_{j=1}^{i-1} \alpha_j}.$$

These values can be written as follows:

$$S = g_2^{\alpha_W+\sum_{j=1}^{i-1} \alpha_j} \left(u' \prod_{j=1}^{l} u_j^{m_j^{(h)}} \right)^r (R_i)^{\sum_{j=1}^{i}(jt_j+v_j)},$$
$$R_i = g_1^r,$$

where r is a secret random number. $\mathcal{B}$ outputs (S_i, R_i) as an ordered multisignature for pk^*.

Output: Also this step is almost the same as that in Ref. [18]. After $\mathcal{A}$ outputs a forgery $\sigma^* = (S^*, R^*)$, on a message m^* in ψ_n^*, $\mathcal{B}$ can extract a forgery $\sigma_W^* = (S_W^*, R_W^*)$ of the Waters signature from the $\mathcal{A}$'s output. Let the target signer be i^*-th signer in $\psi_{i^*}^*$. Then, $\mathcal{B}$ sets $R_W^* = R^*$ and extract S_W^* as follows:

$$S_W^* = \frac{S^*}{g_2^{\sum_{j=1 \wedge j \neq i^*}^{n} \alpha_j} (R)^{\sum_{j=1}^{n}(jt_j+v_j)}}.$$

Finally, $\mathcal{B}$ outputs (S_W^*, R_W^*) as a forgery for the Waters signature scheme. The success probability ϵ_W and an execution time t_W of $\mathcal{B}$ are given similarly as that of the method in Ref. [18]. There is no event in which $\mathcal{B}$ aborts the simulation. Therefore, $\epsilon_W = \epsilon$ and $q_{W_s} = q_s$ hold. The execution time t_W is given from that of $\mathcal{A}$ plus two exponentiation computations and two bilinear map computations for **Certification Query**, two exponentiation computations for **Signing Query** and two exponentiation computation for the final step. Therefore, $t_W = t + (2q_c + 2q_s + 2)t_e + 2q_c t_p$, where t_e is the computational cost for the exponentiation. □

Here, We note the following theorem [29].

Theorem 6. The Waters signature scheme is (t, q, ϵ)-secure if (t', ϵ')-CDH assumption holds, where $t' = t$ and $\epsilon' = \frac{\epsilon}{16(l+1)q}$.

Proof (Sketch). The proof is given in Ref. [29]. □

This theorem implies that, when the proposed scheme is broken, we can construct an algorithm to solve the CDH problem with (t', ϵ'), where $t' = t+(3q_c+2q_s+2)t_e+2q_c t_p$ and $\epsilon' = \frac{\epsilon}{16(l+1)q_s}$.

(**case 2**): This proof is based on the proof in Ref. [6].

Lemma 7. The proposed ordered multisignature scheme is $(t, q_c, q_s, l, n, \epsilon)$-secure if (t'', ϵ'')-CDH assumption holds, where $t'' = t + (3q_c + 9q_s + 2)\, t_e + 2q_c t_p$ and $\epsilon'' = \frac{\epsilon}{e(q_s-1)}$.

Proof. In order to solve the CDH problem, $\mathcal{B}$ interacts with $\mathcal{A}$ as follows:

Initial Phase: Given a challenge value (g, g^a, g^b) for CDH problem and a pairing parameter $(p, \mathbb{G}, \mathbb{G}_T, \boldsymbol{e})$, $\mathcal{B}$ sets $\mathcal{L} = \emptyset$, and generates ℓ-dimensional vector $\boldsymbol{x_i} \leftarrow \mathbb{Z}_p^l$ and $x' \leftarrow \mathbb{Z}_p$. For a message m, we define polynomials $F(m) = x' + \sum_{i=1}^{\ell} x_i m_i$, where m_i corresponds to i-th bit in m. $\mathcal{B}$ also sets $u' = g^{x'}$ and $u_i = g^{x_i}$ as each generator for public parameter, i.e., $(u' \prod_{j=1}^{l} u_j^{m_j}) = g^{F(m)}$. $\mathcal{B}$ sets $g_1 = g$ and $g_2 = g^b$. Finally, $\mathcal{B}$ generates random numbers $k^* \leftarrow [1, n]$, $(t^*, v^*) \leftarrow \mathbb{Z}_p^*$, and then sets $T^* = (g^a)^{t^*}$, $V^* = (g^a)^{-t^*k^*} g^{v^*}$ and $A^* = g^a$ as the public key pk^* of the target signer. This means that $\mathcal{B}$ sets implicitly g_2^a as the target signer's secret key. Then, $\mathcal{B}$ runs $\mathcal{A}$ with $(p, \mathbb{G}, \mathbb{G}_T, \boldsymbol{e}, g_1, g_2, u', u_1, \cdots, u_\ell, A^*, T^*, V^*)$.

Certification Query: This step is the same as that of (**case 1**).

Signing Query: This step is almost the same as that in Ref. [6]. $\mathcal{B}$ checks that $i^{(h)} = k^*$ holds, where $i^{(h)}$ is the position of the target signer for the h-th query. If so, $\mathcal{B}$ aborts the process. Otherwise, $\mathcal{B}$ discards a signature given from $\mathcal{A}$. Then, $\mathcal{B}$ generates a random number $r \leftarrow \mathbb{Z}_p$ and computes as follows:

$$\begin{aligned} S_i &= (g^b)^{-\frac{v^*}{t^*(i-k^*)}} ((g^a)^{it^*} (g^a)^{-t^*k^*} \times g^{v^*})^r \\ &\quad \times (R_i)^{F(m^{(h)})+\sum_{j=1}^{i-1}(jt_j+v_j)} g_2^{\sum_{j=1}^{i-1} \alpha_j}, \\ R_i &= g^r (g^b)^{-\frac{1}{t^*(i-k^*)}}. \end{aligned}$$

Let $r' := r - \frac{b}{t^*(i-k^*)}$, and firstly the following equations hold from **Initial Phase** as a part of S_i:

$$\begin{aligned} S' &= (g^b)^{-\frac{v^*}{t^*(i-k^*)}} g^{ab} g^{-ab\frac{t^*(i-k^*)}{t^*(i-k^*)}} \left((g^a)^{it^*} (g^a)^{-t^*k^*} g^{v^*}\right)^r \\ &= g^{ab} \left(g^{v^*} g^{at^*(i-k^*)}\right)^{-\frac{b}{t^*(i-k^*)}} \left((g^a)^{it^*} (g^a)^{-t^*k^*} g^{v^*}\right)^r \\ &= g^{ab} \left((g^a)^{it^*} (g^a)^{-t^*k^*} g^{v^*}\right)^{r-\frac{b}{t^*(i-k^*)}} = g^{ab} \left(T^{*i} V^*\right)^{r'}. \end{aligned}$$

Then, the following equation holds:

$$\begin{aligned} S_i &= S'(R_i)^{F(m^{(h)})+\sum_{j=1}^{i-1}(jt_j+v_j)} g_2^{\sum_{j=1}^{i-1} \alpha_j}, \\ &= g_2^{a+\sum_{j=1}^{i-1} \alpha_j} \left(u' \prod_{j=1}^{l} u_j^{m_j^{(h)}}\right)^{r'} \left(\prod_{j=1}^{i} T_j^j V_j\right)^{r'}, \\ R_i &= g_1^{r'}. \end{aligned}$$

$\mathcal{B}$ returns (S_i, R_i) as an ordered multisignature for pk^*.

Output: $\mathcal{A}$ outputs a forgery $\sigma^* = (S^*, R^*)$ on a message m^* in ψ_n^*, and let the target signer be i^*-th signer in ψ_n^* where $\psi_{i^*}^*$ is extracted from ψ_n^*. If $i^* = k^*$, $\mathcal{B}$ can solve the CDH problem since the forgery can be written as follows:

$$\begin{aligned} \boldsymbol{e}(S^*, g_1) &= \boldsymbol{e}\left(g^{ab} \prod_{i=1 \wedge i \neq i^*}^{n} g_2^{\alpha_i} \left(u' \prod_{j=1}^{\ell} u_j^{m_j}\right)^r \left(\prod_{j=1}^{n} T_j^j V_j\right)^r, g_1\right) \\ \Rightarrow S^* &= g^{ab} \prod_{i=1 \wedge i \neq i^*}^{n} g_2^{\alpha_i} \left(u' \prod_{j=1}^{l} u_j^{m_j}\right)^r \left(\prod_{j=1 \wedge j \neq k^*}^{n} T_j^j V_j\right)^r (g^{v^*})^r, \end{aligned}$$

where r is a secret value in the exponent of R^*. Then, $\mathcal{B}$ can solve the CDH problem by the following computation:

$$g^{ab} = \frac{S^*}{(R^*)^{F(m)+\sum_{j=1 \wedge j \neq k^*}(jt_j+v_j)+v^*} g_2^{\sum_{j=1 \wedge j \neq k^*}^{n} \alpha_j}}.$$

The success probability of $\mathcal{B}$ is $\epsilon'' = \epsilon \Pr[(\bigwedge_{j=1}^{q_s} i^{(j)} \neq k^*) \wedge i^* = k^*)] = \epsilon\left(1 - \frac{1}{n}\right)^{q_s} \cdot \frac{1}{n}$. Here, we analyze $\left(1 - \frac{1}{n}\right)^{q_s} \frac{1}{n}$ similarly as the proof in Ref. [6]. Then, $\epsilon'' = \epsilon \cdot \frac{1}{e(q_s-1)}$ holds from the derived function with respect to n and its extremum. The $\mathcal{B}$'s execution time t' is that of $\mathcal{A}$ plus three exponentiation computations and two bilinear maps for **Certification Query**, nine exponentiation computations for **Signing Query** and two exponentiation for the final step. Therefore, $t'' = t + (3q_c + 9q_s + 2)t_e + 2q_c t_p$ holds. □

Analysis of Whole Probability: In order to success for each case, $\mathcal{B}$ needs that either one of the following events occurs: $\mathcal{B}$ chooses (**case 1**) and $\mathcal{A}$'s output is for (**case 1**); $\mathcal{B}$ chooses (**case 2**) and $\mathcal{A}$'s output is for (**case 2**). Here, let the probability that $\mathcal{B}$ chooses (**case 1**) be β and the probability that $\mathcal{A}$'s output is for (**case 1**) be α. From Lemma 5, Theorem 6 and Lemma 7,

Table 1 Evaluation of the schemes: We denote by ℓ the message length, by λ the number of chunks, by $\mathcal{P}$ the computational cost of the bilinear map, by $\mathcal{E}$ the computational cost of exponentiation, and by $\mathcal{L}(p)$ the binary length of p.

Schemes	Computational Cost for i-th Signer	Computational Cost for Verifier	Signature Size	Public Key Size
Ahn et al. [1]	$\left(\frac{\ell}{\lambda}+5\right)\mathcal{E}$	$\left(\frac{\ell}{\lambda}+3\right)\mathcal{P}+2\mathcal{E}$	$2\mathcal{L}(p)$	$\mathcal{L}(p)$
Lu et al. [18]	$5\mathcal{E}$	$3\mathcal{P}$	$2\mathcal{L}(p)$	$(\ell+2)\mathcal{L}(p)$
Our Scheme	$(i+4)\mathcal{E}$	$4\mathcal{P}+n\mathcal{E}$	$2\mathcal{L}(p)$	$3\mathcal{L}(p)$

Table 2 Evaluation of the schemes: We compare concrete values for Table 1. For 128-bit security, i.e., $l = 256$, and $n = 20$, each value is given as follows, where $\lambda = 32$ [21], the time per one $\mathcal{P}$ is 6.398 msec and the time per one $\mathcal{E}$ is 0.6836 msec by using TEPLA [14].

Schemes	Computational Time for 20-th Signer (msec)	Computational Time for Verifier with 20 Signers (msec)	Signature Size (bits)	Public Key Size (bits)
Ahn et al. [1]	8.8868	71.7452	1,024	512
Lu et al. [18]	3.418	19.194	1,024	263,168
Our Scheme	16.40	39.264	1,536	1,536

Table 3 Evaluation of the schemes: We compare the total costs of Table 1 with 20 signers. The following values are obtained by repeating the computation of Table 2 from $i = 1$ to $i = 20$. The total computational time is the computational time for the signers plus that of the verifiers, and the total communication size is the signature size plus the public key size for 20 signers. ○ means that the total packet size in the scheme is smaller than the packet limitation of BGP.

Schemes	Total Computational Time for 20 Signers (msec)	Total Communication Cost for 20 Signers (bytes)	Total Communication Cost is Less than Packet Limitation (4,096 bytes)
Ahn et al. [1]	1,612.64	1,408	○
Lu et al. [18]	452.24	658,048	
Our Scheme	853.640	4,032	○

$\epsilon_{CDH} = \alpha\cdot\beta\cdot\frac{\epsilon}{16(l+1)q_s}+(1-\alpha)\cdot(1-\beta)\frac{1}{e(q_s-1)}\cdot\epsilon$ holds. In order to be a complete proof, we analysis values of α and β. Let $f(\alpha,\beta)$ denote $\alpha\cdot\beta\cdot\frac{\epsilon}{a}+(1-\alpha)\cdot(1-\beta)\frac{\epsilon}{b}$, where $a = 16(l+1)q_s$ and $b = e(q_s-1)$ as constants shortly. Then, its derived function with respect to α is computed as $\frac{\partial f(\alpha,\beta)}{\partial\alpha} = \beta\cdot\frac{\epsilon}{a}+(-1)(1-\beta)\frac{\epsilon}{b}$. The function has an extremum at $\beta = \frac{a}{a+b}$. Therefore, when $\mathcal{B}$ sets $\beta = \frac{a}{a+b}$, the probability can be obtained as $\epsilon_{CDH} = \frac{\epsilon}{16(l+1)q_s+e(q_s-1)}$. The computational time t_{CDH} can be obtained as the larger value between t' and t''. Therefore, $t_{CDH} = \max\{t',t''\} = t'' = t+(3q_c+9q_s+2)t_e+2q_ct_e$ holds. □

6. Evaluation

We compare the performance of the proposed scheme with the existing ordered multisignature schemes, which are straightforwardly obtained from the aggregate signature schemes, in the standard model [1], [18] with respect to the signing cost, the verification cost, the signature size, and the size of the public key [*4]. Here, for the signing and the verification costs, we compare these schemes in terms of the number of computations of the bilinear maps and the number of exponentiation computations.

As shown in **Table 1**, in comparison with Refs. [1], [18], the number of computations of the bilinear maps for the verification cost and the size of the public keys in our scheme are independent of the length of the message. Although for our proposed scheme, the number of exponentiation is linear with respect to the number of signers, in general, the cost of the exponentiation much less than that of computations of the bilinear maps. In addition, the costs of the exponentiation computations can be reduced via the Pippenger algorithm [4].

For instance, we compare concrete values in **Table 2** in the case of 20 signers, which is a typical number of signers in the S-BGP (see Section 7). The values in Table 2 are given relative to the benchmark of TEPLA [14].

Table 3 shows total costs of the schemes for 20 signers in the BGP application described in Section 7. The total communication costs of the scheme by Lu et al. are 657,048 bytes, which is larger than the packet limitation 4,096 bytes of BGP UPDATE [16]. Furthermore, in comparison with Ahn et al.'s scheme, as described in **Fig. 1**, the computational time of our scheme is roughly half. Thus, our proposed scheme is the most practical in the existing CDH-based schemes.

7. Applications

One of the possible applications of an ordered multisignature scheme is an improvement of *secure-border gateway protocol* (*S-BGP*) [16], which is a variant of BGP where each AS sign path information corresponding to the policy. Whereas S-BGP enforces ASes to authenticate their own paths, Feamster et al. [12] pointed out that S-BGP provides only a weak security: in particular, an AS router should reject the packets from invalid sources. They called such a new capability the data-plane security, and it is a more advanced primitive where the validity of the actual forwarding path for data packets can be guaranteed. Namely, an advantage of the data-plane security is that packets are certainly forwarded along with the authenticated path information.

According to Boldyreva et al. [6], ordered multisignatures seem to be suitable for the data-plane security: more specifically, in order to provide the data-plane security, ASes can sign the data packets by using an ordered multisignature scheme where the signing order represents the actual forwarding path and the packets themselves correspond to messages. More precisely, each

[*4] Actually, the form of the evaluation is different from that in Ref. [18]. We evaluated an optimized scheme whose form of the public key is the same as that of our proposed scheme.

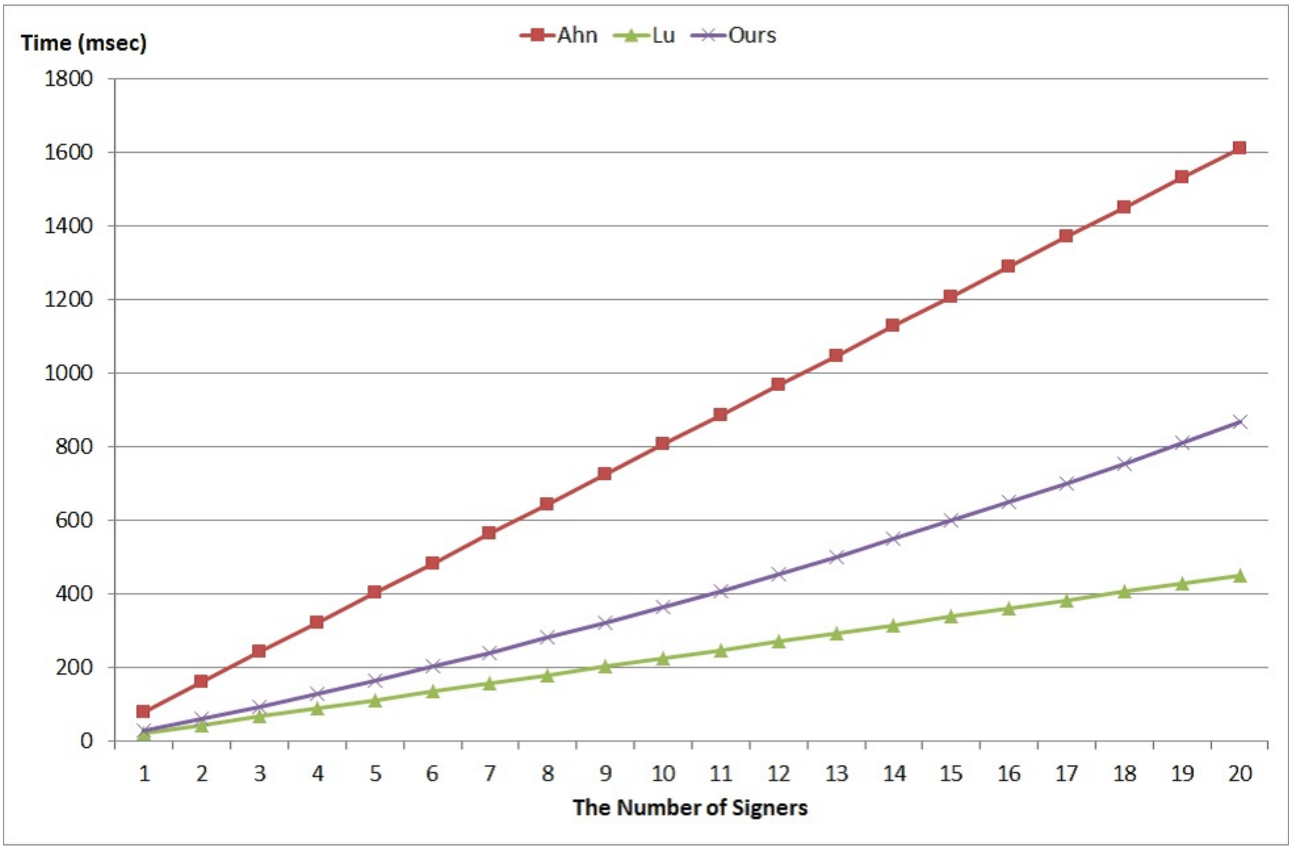

Fig. 1 Evaluation of the schemes.

packet includes a source address and a destination address, and these information in the packet is signed by egress routers of ASes forwarding it. The egress routers insert their signatures and public key information into the data packets. Ingress routers receiving the packets forward only packets with valid ordered multisignatures that followed an authenticated path, where the routers check that the validity of the signatures with the their own path information as the signing order. If the verification algorithm rejects the signatures, the ingress routers will eliminate the packets. Finally, the origin AS of the packets verifies that the packet actually took an authenticated path to reach its destination. In comparison with standard signatures such as ECDSA or standard multisignatures, by virtues of the order verifiability and an aggregation of the signatures, ordered multisignatures allow reduction of the computational overloads and verification of records of the packet forwarding.

This application is different from the traditional S-BGP. The data-plane security guarantees the validity of the actual path which forwards the packets while the traditional S-BGP guarantees only the validity of the path information in routers. The cost for signing the data packets is larger than that for the paths, because there is a far larger amount of data in comparison to the path information.

The application described above can be achieved by the proposed scheme. The scheme by Lu et al. is that the size of the public key is larger than the limitation of the packet size of the BGP, which is 4,096 bytes. Meanwhile, the scheme by Ahn et al. requires a large amount of the computational time, whose cost is about twofold relatively to our scheme. In other words, the fastest scheme which is available for securing BGP is our proposed scheme.

8. Conclusion

In this paper, we proposed an ordered multisignature scheme without the random oracles. Most of the existing ordered multisignature schemes adopt the random oracle model to analyze security, and, to the best of our knowledge, the existing CDH-based schemes are impractical for obtaining the data-plane security of the S-BGP, which is the main application. Also, to the best of our knowledge, our scheme is the first CDH-based scheme that achieves all of the following conditions: rigorous security analysis in the standard model under the moderate attack model, a fixed number of computations of the bilinear maps, and a fixed size to the public key with respect to the length of the message. These results mean that our scheme is the best to use for the application described in Section 7. We plan to discuss a scheme not requiring the moderate attack model and to extend the method to allow for more complicated structures of signers, e.g., a hierarchical structure. We also plan to propose an identity-based ordered multisignature scheme, similar to the schemes in Ref. [28] in the standard model.

Acknowledgments Part of this research is supported by JSPS A3 Foresight Program. The first author is also supported by Support Center for Advanced Telecommunications Technology Research. We would like to appreciate their great supports. We would also like to appreciate anonymous reviewers and Shin-Akarui-Angou-Benkyou-Kai for their valuable comments.

References

[1] Ahn, J.H., Green, M. and Hohenberger, S.: Synchronized Aggregate Signatures: New Definitions, Constructions and Applications, *Proc. 17th ACM Conference on Computer and Communications Security*, pp.473–484, ACM (2010), (online), available from ⟨http://eprint.iacr.org/2010/422⟩, Full paper is available in Cryptology ePrint Archive.

[2] Bellare, M., Namprempre, C. and Neven, G.: Unrestricted Aggregate Signatures, *Proc. 34th International Colloquium on Automata, Languages and Programming* (*ICALP 2007*), Lecture Notes in Computer Science, Vol.4596, pp.411–422, Springer-Verlag (2007).

[3] Bellare, M. and Rogaway, P.: Random Oracles are Practical: A Paradigm for Designing Efficient Protocols, *Proc. 1st ACM Conference on Computer and Communications Security* (*CCS 1993*), pp.62–73, ACM (1993).

[4] Bernstein, D.J.: Pippenger's Exponentiation Algorithm (2002), available from ⟨http://cr.yp.to/papers/pippenger.pdf⟩.

[5] Boldyreva, A.: Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme, *Proc. 6th International Workshop on Practice and Theory in Public Key Cryptography* (*PKC 2003*), Lecture Notes in Computer Science, Vol.2567, pp.31–46, Springer-Verlag (2003).

[6] Boldyreva, A., Gentry, C., O'Neill, A. and Yum, D.H.: Ordered Multisignatures and Identity-Based Sequential Aggregate Signatures, with Applications to Secure Routing (Extended Abstract), *Proc. 14th ACM Conference on Computer and Communication Security* (*CCS 2007*), pp.276–285, ACM (2007).

[7] Boldyreva, A. and Lychev, R.: Provable Security of S-BGP and Other Path Vector Protocols: Model, Analysis and Extensions, *Proc. 19th ACM Conference on Computer and Communications Security* (*CCS 2012*), pp.541–552, ACM (2012), (online), available from ⟨http://eprint.iacr.org/2013/017⟩. Full paper is available in Cryptology ePrint Archive.

[8] Boneh, D., Gentry, C., Lynn, B. and Shacham, H.: Aggregate and Verifiably Encrypted Signatures from Bilinear Maps, *Proc. 22th International Conference on the Theory and Applications of Cryptographic Techniques* (*EUROCRYPT 2003*), Lecture Notes in Computer Science, Vol.2656, pp.416–432, Springer-Verlag (2003).

[9] Brogle, K., Goldberg, S. and Reyzin, L.: Sequential Aggregate Signatures with Lazy Verification from Trapdoor Permutations, *Proc. 18th International Conference on the Theory and Application of Cryptology and Information Security* (*ASIACRYPT 2012*), Lecture Notes in Computer Science, Vol.7658, pp.663–680 (2012), (online), available from ⟨http://eprint.iacr.org/2011/222⟩, Full paper is available in Cryptology ePrint Archive.

[10] Canetti, R., Goldreich, O. and Halevi, S.: The Random Oracle Methodology, Revisited, *J. ACM*, Vol.51, No.4, pp.557–594 (2004).

[11] Doi, H., Mambo, M. and Okamoto, E.: Multisignature Schemes Using Structured Group ID, *Technical Report of IEICE*, Vol.98, No.464, pp.43–48, IEICE (1998).

[12] Feamster, N., Balakrishnan, H. and Rexford, J.: Some Foundational Problems in Interdomain Routing, *Proc. 3rd Workshop on Hot Topics in Networking* (*HotNets-3*), ACM (2004), (online), available from ⟨http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.132.3571&rep=rep1&type=pdf⟩.

[13] Hofheinz, D. and Kiltz, E.: Programmable Hash Functions and Their Applications, *Proc. 28th Annual International Cryptology Conference* (*CRYPTO 2008*), Lecture Notes in Computer Science, Vol.5157, pp.21–38, Springer-Verlag (2008).

[14] Ishii, K., Saito, K., Kanaoka, A., Kanayama, N. and Okamoto, E.: General Purpose C Library for Pairing Cryptography, *Proc. 30th Symposium on Cryptography and Information Security* (*SCIS 2013*), IEICE (2013), in Japanese.

[15] Itakura, K. and Nakamura, K.: A Public-key Cryptosystem Suitable for Digital Multi-signatures, *NEC Research and Development*, Vol.71, pp.1–8 (1983).

[16] Kent, S., Lynn, C. and Seo, K.: Secure Border Gateway Protocol, *IEEE Journal of Selected Areas in Communications*, Vol.18, No.4, pp.582–592 (2000).

[17] Lee, K., Lee, D.H. and Yung, M.: Sequential Aggregate Signatures with Short Public Keys: Design, Analysis and Implementation Studies, *The 16th International Conference on Practice and Theory in Public-Key Cryptography* (*PKC 2013*), Lecture Notes in Computer Science, Vol.7778, pp.423–442, Springer-Verlag (2013).

[18] Lu, S., Ostrovsky, R., Sahai, A., Shacham, H. and Waters, B.: Sequential Aggregate Signatures and Multisignatures without Random Oracle, *Proc. 25th Theory and Applications of Cryptographic Techniques* (*EUROCRYPT 2006*), Lecture Notes in Computer Science, Vol.4004, pp.465–485, Springer-Verlag (2006).

[19] Lysyanskaya, A., Micali, S., Reyzin, L. and Shacham, H.: Sequential Aggregate Signatures from Trapdoor Permutations, *Proc. 23rd Annual International Conference on the Theory and Applications of Cryptographic Techniques* (*EUROCRYPT 2004*), Lecture Notes in Computer Science, Vol.3027, pp.74–90, Springer-Verlag (2004).

[20] Mitomi, S. and Miyaji, A.: A General Model of Multisignature Schemes with Message Flexibility, *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, Vol.E84-A, No.10, pp.2488–2499 (2001).

[21] Naccache, D.: Secure and Practical Identity-Based Encryption (2005), Full paper is available in Cryptology ePrint Archive, available from ⟨http://eprint.iacr.org/2005/369⟩.

[22] Neven, G.: Efficient Sequential Aggregate Signed Data, *IEEE Trans. Inf. Theory*, Vol.57, No.3, pp.1803–1815 (2011).

[23] Ohta, K. and Okamoto, T.: Multi-Signature Schemes Secure against Active Insider Attacks, *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, Vol.E82-A, No.1, pp.21–31 (1999).

[24] Rekhter, Y. and Li, T.: A Border Gateway Protocol 4 (BGP-4) (1995), RFC 1771, available from ⟨http://www.ietf.org/rfc/rfc1771.txt⟩.

[25] Rückert, M. and Schröder, D.: Aggregate and Verifiably Encrypted Signatures from Multilinear Maps without Random Oracles, *Proc. 3rd International Conference on Information Security and Assurance* (*ISA 2009*), Lecture Notes in Computer Science, Vol.5576, pp.750–759, Springer-Verlag (2009).

[26] Schröder, D.: How to Aggregate the CL Signature Scheme, *Proc. 16th European Symposium on Research in Computer Security*, Lecture Notes in Computer Science, Vol.6879, pp.298–314, Springer-Verlag (2011).

[27] Sriram, K., Borchert, O., Kim, O., Cooper, D. and Montgomery, D.: RIB Size Estimation for BGPSEC (2011), available from ⟨http://www.antd.nist.gov/˜ksriram/BGPSEC_RIB_Estimation.pdf⟩.

[28] Wang, L., Okamoto, E., Miao, Y., Okamoto, T. and Doi, H.: An ID-SP-M4M Scheme and Its Security Analysis, *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, Vol.E90-A, No.1, pp.91–100 (2007).

[29] Waters, B.: Efficient Identity-Based Encryption without Random Oracles, *Proc. 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques* (*EUROCRYPT 2005*), Lecture Notes in Computer Science, Vol.3494, pp.114–127, Springer-Verlag (2005).

[30] Yanai, N., Chida, E., Mambo, M. and Okamoto, E.: A Secure Ordered Multisignature without Random Oracles, *Proc. Computer Security Symposium* (*CSS 2012*), Vol.2012, No.3, pp.293–300, IPSJ (2012).

Editor's Recommendation

The authors propose an efficient ordered multi-signature scheme, whose public key size is independent of the number of signers and the length of message. Multi-signature schemes have been investigated in cryptography for a long time, but the proposed scheme is the most efficient one which is proved secure under the standard model based on CDH assumption. This result is a milestone in the research of multi-signatures, and we expect that this paper provides further research developments in cryptography.

(Program Chair of Computer Security Symposium 2012,
Tsuyoshi Takagi)

Naoto Yanai received B.Eng. degree in electrical engineering from Ichinoseki National College of Technology, Japan, in 2009 and M.S.Eng. in graduate school of systems and information engineering from Univeirsity of Tsukuba, Japan, in 2011. He has recently joined Dr. course in systems and information engineering in University of Tsukuba, Japan.

Eikoh Chida received his B.Eng. degree in information engineering and M.S. and Ph.D. degrees in information science from Tohoku University, Japan, in 1993, 1995 and 1998, respectively. From 1995 to 1998, he was a member of Research Fellow of the Japan Society for the promotion of Science. He joined Tohoku University in 1998 as an assistant professor of both Education Center for Information Processing and Graduate School of Information Sciences. He is currently an associate Professor of the Department of Electrical Engineering, Ichinoseki National College of Technology. His research interests include cryptology and related mathematics.

Masahiro Mambo received his B.Eng. degree from Kanazawa University, Japan, in 1988 and M.S.Eng. and Dr.Eng. degrees in electronic engineering from Tokyo Institute of Technology, Japan in 1990 and 1993, respectively. After working at Japan Advanced Institute of Science and Technology, JAIST, at Tohoku University and at University of Tsukuba as assistant, associate and associate professor, respectively, he joined Kanazawa University in 2011. He is currently a professor of Faculty of Electrical and Computer Engineering, Institute of Science and Engineering. His research interests include information security, software protection and privacy protection.

Eiji Okamoto received his B.S., M.S. and Ph.D. degrees in electronics engineering from the Tokyo Institute of Technology in 1973, 1975 and 1978, respectively. He worked and studied communication theory and cryptography for NEC central research laboratories since 1978. In 1991 he became a professor at Japan Advanced Institute of Science and Technology, then at Toho University. Now he is a professor at Faculty of Engineering, Information and Systems, University of Tsukuba. His research interests are cryptography and information security. He is a coeditor-in-chief of Internatinal Journal of Information Security, and a member of IEEE and ACM.