スタンダードモデルにおける順序検証型多重署名方式

矢内 直人 †　　千田 栄幸 ‡　　満保 雅浩 ††　　岡本 栄司 †

† 筑波大学 システム情報工学研究科
‡ 一関工業高等専門学校 電気工学科
†† 金沢大学 自然科学研究科

あらまし 順序検証型多重署名は電子データの正当性とその処理順序の両方を検証可能な署名方式であり, その既存方式の多くははランダムオラクルモデルにおいて提案されている. しかしながら, ランダムオラクルの実装の困難性を考えた場合, 安全性証明はランダムオラクルを用いないスタンダードモデルにて行われることが望ましい. スタンダードモデルでの順序検証型多重署名方式を構成する最も簡単な方式は, スタンダードモデルにおける既存のアグリゲート署名から構成することであり, 既存のアプローチとして Ahn らと Lu らの CDH ベースアグリゲート署名方式があるが, これらの方式はペアリングの演算回数と公開鍵のサイズがメッセージのハッシュ長に依存するという非効率な面を持つ. それゆえに, 本論文ではペアリング演算回数と公開鍵のサイズがメッセージのハッシュ長に独立した CDH ベース順序検証型多重署名方式を提案する. また, Boldyreva らの方式の安全性の不備を指摘し, それに伴い設定される適度な攻撃モデルにおいて提案方式の安全性を示す.

A Secure Ordered Multisignature without Random Oracles

Naoto Yanai†　　Eikoh Chida‡　　Masahiro Mambo††　　Eiji Okamoto†

†Graduate School of Systems and Information Engineering University of Tsukuba
‡Department of Electrical and Computer Engineering Ichinoseki National College of Technology
††Institute of Science and Engineering Kanazawa University

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 an aggregate signature scheme by Ahn et al. and an aggregate signature scheme by Lu et al., both of which are based on CDH problem, but these schemes are inefficient in the sense that its computational cost of pairing computation and the size 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 whose computational cost for pairing computation and the size of public key are independent of the length of (a hash value of) the message. We also point out a bug of the scheme by Boldyreva et al., and analyze the security of our scheme under a moderate attack model along with fixing the bug.

1 Introduction

1.1 Background

Multisignature is a digital signature generated by multiple signers on an electronic document [8], and the data size of a multisignature is designed to be smaller than that of a trivial signature consisting of all individual signatures generated by all the associated signers. A multisignature scheme is called *ordered multisignature scheme* [6] if each signer signs in turn and a verifier verifies both validity of the message and its signing order. According to [3] one may imagine that an ordered multisignature scheme can be simply constructed from aggregate signature scheme [4]. Particularly,

each signer sets concatenation of his/her document and his/her position in the signing group as a message and signs it. We call such an approach *simple ordered multisignature scheme*. However, to the best of our knowledge, the security of simple ordered multisignature schemes has never been proven while such a construction seems to be secure. Currently, many ordered multisignature schemes including simple ordered multisignature schemes are proposed.

Although the security of the most ordered multisignature schemes have been proven in the random oracle model [2], Canetti et al. showed a negative result [5] that there exist signature and encryption schemes, which are secure in the random oracle model but become insecure by an implementation of the random oracles. This result implies that the security should be proven without random oracles, and such a model is called *standard model*.

In this approach, Ahn et al. [1] and Lu et al. [11] have respectively proposed aggregate signature schemes based on CDH problem, and we can also construct simple ordered multisignature schemes in the standard model by applying these schemes. They have a problem of the efficiency; both the pairing computation in the scheme by Ahn et al. and the size of the public key in the scheme by Lu et al. increase linearly in the length of the message. (Strictly speaking, in the length of a hashed value of the message because messages are hashed before signing. Hereafter, it is simply called the length of the message). Additionally, the security of simple ordered multisignature schemes may be unclear as described above.

As described in section 1.2, according to Boldyreva et al. [3] one of major applications of ordered multisignature scheme is the improvement of secure-BGP [10], and its main problem is the limitation of computational power and memories of routers. Thus, in this paper we propose a CDH based ordered multisignature scheme in the standard model, whose pairing computation and the public key size are independent of the length of the message, and rigorously analyze the security in comparison to the simple ordered multisignature schemes. We also point out a bug in [3], and revise an attack model. This paper is a revised version of the paper [16] proposed in SCIS 2011, and we fix a bug of the security proof.

1.2 Application

One of main applications of ordered multisignature scheme is the improvement of *S-BGP (secure-border gateway protocol)*, which is a border-gateway-protocol[12]. S-BGP enforces autonomous systems (ASes) to authenticate paths and to send data packet via the authenticated paths. Boldyreva et al.[3] proposed that ASes can sign the data packets by utilizing ordered multisignature scheme, and this function is called *data plane* security in contrast to the traditional S-BGP. Egress routers of each AS sign the data packets, and the ingress routers of the next AS verify the validity. This application is different from the traditional S-BGP in the sense that each AS router authenticates the data packets instead of paths. Unfortunately, the size of signatures for the packets becomes larger than that of the signatures for the paths.

Namely, the data size of S-BGP with the scheme by Lu et al., including the public keys, may become a few giga bytes, and this traffic is much heavy. In addition, S-BGP with the scheme by Ahn et al. may cause a long delay because the number of pairing computation in the verification depends on the message length. Thus, achieving data plane security by utilizing the existing schemes may be impractical. Since the computational cost and the size of public key in our proposed scheme are fixed with respect to the length of the message, we expect that the scheme can be applied to a scheme achieving the data plane security.

2 Preliminaries

2.1 Notations

First, we introduce notations used in this paper. Let the number of signers be n. For simplicity, we denote by ID_i the i-th signer if the notation does not cause any confusion. We also 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 a signature generated by a signer ID_i, by pk_i his/her public key, by sk_i his/her secret key and by $a \parallel b$ a concatenation of elements a and b, where the concatenation can be easily divided into original elements a and b. We define $\psi_i := ID_1 \parallel \cdots \parallel ID_i$ as the signing order from the first signer ID_1 to i-th signer ID_i and denote by $|\psi_i|$ the number of signers in ψ_i.

2.2 Pairings

Let $\mathbb{G}$ and $\mathbb{G}_T$ be groups. Then, we define bilinear maps and bilinear groups as follows:

Definition 1 (bilinear maps)**.** A bilinear map $\boldsymbol{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a map such that the following conditions hold: $\mathbb{G}$ and $\mathbb{G}_T$ are groups of the same prime order p; 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 the conditions hold, and assume that discrete logarithm problem (DLP) in bilinear groups is hard. We call such a parameter $(p, \mathbb{G}, \mathbb{G}_T, \boldsymbol{e})$ *pairing parameter*.

2.3 Security Assumption

Computational Diffie-Hellman (CDH) assumption is defined as follows.

Definition 2 (CDH problem)**.** We define the computational Diffie-Hellman (CDH) problem in bilinear groups with a security parameter 1^k as the problem of, for a given $(g, g^a, g^b) \in \mathbb{G}$ with uniformly random $a, b \in \mathbb{Z}_p$ as input, computing g^{ab}, where p is a prime order of $\mathbb{G}$.

Definition 3 ((t, ϵ)-CDH assumption)**.** 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 probability greater than ϵ with the execution time t.

2.4 Related Work

Among all the existing schemes, including aggregate signature schemes, the schemes in the standard model are proposed in [1, 7, 11, 13, 14]. Unfortunately among them, it seems that the security proof in [7] is wrong. Hence, the secure schemes are only in [1, 11, 13, 14]. To achieve the higher security, the scheme should be based on more general problem such as CDH problem. Among them, CDH-based schemes are only in [1, 11]. However, the scheme in [1] has the problem for the computational cost and the scheme in [11] has the problem about the storage of the public keys.

3 Ordered Multisignature

In this section, we explain a general construction of ordered multisignature scheme, the bug in [3], the security and a main idea.

3.1 General Construction

Ordered multisignature scheme consists of the following algorithms.

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

Key Generation: Given *para* and signer's ID information ID_i, 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' by the previous signer and the signing order ψ_{i-1}, generate a signature σ. Finally, set $\psi_i = \psi_{i-1} \parallel ID_i$ and then output the signature σ on m in ψ_i.

Verification: Given m, σ, ψ_n and $\{pk_i\}_{i=1}^n$, output *accept* or *reject*.

3.2 Bug of the Scheme in [3]

The signature equation in [3] can be written as $S = H(m)^{\sum_{j=1}^{i} \alpha_i} \left(\prod_{j=1}^{i} T_j^j V_j\right)^r$ for any i, and the whole signature is given as a tuple of $(S, R = g^r)$, where secret key of i-th signer is α_i and public key of him/her is (T_i, V_i). In their proof, signers are allowed to attend in multiple positions for one signature generation, e.g. Alice‖Bob‖Alice is allowed as a signing order. However, such multiple appearance of signers induces the following attack. In the case of ψ_3 =Alice‖Bob‖Alice, the signature equation for ψ_3 becomes $S = H(m)^{2\alpha_a+\alpha_b} \left(T_a^4 V_a^2 T_b^2 V_b\right)^r$, where α_a is a secret key of Alice and α_b is that of Bob. Then, Bob who is a malicious signer can forge a signature for another signing order ψ_2 =Bob‖Alice as follows: He/She compute $S^{\frac{1}{2}} H(m)^{\frac{\alpha_b}{2}} (R)^{\frac{v_b}{2}}$, where v_b is his/her own secret key corresponding to V_b. This value is equal to $H(m)^{\alpha_a+\alpha_b} \times \left(T_b^1 V_b T_a^2 V_a\right)^r$, which is accepted on m in ψ_2. This is the bug in [3].

The above attack indicates signers should not attend multiple positions in the scheme by Boldyreva et al. Likewise, we restrict signing queries in this model in order to avoid that the same signer attend at multiple times for one signature generation. Actually, some applications such as S-BGP are performed under

such restriction. S-BGP is designed in a way such that a process loop does not occur and the same router never signs again.

3.3 Security Model

There exists an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ in this model. $\mathcal{C}$ has a certified-key list $\mathcal{L}$ to register users and their own secret keys and public keys, and $\mathcal{A}$ can know all the keys in $\mathcal{L}$ except for the target signer's one given by $\mathcal{C}$. The advantage of $\mathcal{A}$ can be obtained with the probability that $\mathcal{C}$ outputs *accept* in the following game. Hereafter, we denote by $x^{(i)}$ a value of i-th query for all x.

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

Certification Query: $\mathcal{A}$ sends any ID_i, and then $\mathcal{C}$ generates sk_i and its corresponding public key pk_i by **Key Generation**(ID_i) as ID_i's key. Then, he/she provides (sk_i, pk_i) to $\mathcal{A}$ and registers pk_i in $\mathcal{L}$.

Signing Query: For all i, $\mathcal{A}$ generates a signing query $(m^{(h)}, \sigma'^{(h)}, ID^*, \psi_{i-1}^{(h)})$ as h-th query for the target signer ID^*, where the following conditions hold for the query and for all h: **Verification** algorithm outputs *accept*; $\psi_{i-1}^{(h)}$ does not include ID^*; For all ID_j, ID_j in $\psi_{i-1}^{(h)}$ is included in $\mathcal{L}$; Each ID_j does not appear more than once in $\psi_{i-1}^{(h)}$; $|\psi_{i-1}^{(h)}| < n$. Given $(m^{(h)}, \sigma'^{(h)}, ID^*, \psi_{i-1}^{(h)})$ by $\mathcal{A}$, $\mathcal{C}$ runs **Signing** $(sk_i, pk_i, m^{(h)}, \sigma'^{(h)}, \psi_i^{(h)})$, and obtains σ and $\psi_i^{(j)} = \psi_{i-1}^{(j)} \parallel ID^*$. Finally, $\mathcal{C}$ returns $\sigma^{(j)}$ on $m^{(j)}$ in $\psi_i^{(j)}$.

Output: After iterations of the steps described above, $\mathcal{A}$ outputs a forgery $(m^*, \sigma^*, \psi_n^*)$. Here, let the target signer be i^*-th signer in ψ_n^*, and the following conditions hold for the forgery: **Verification**$(m, \sigma, \psi_n, \{pk_i\}_{i=1}^n)$ outputs *accept*; $m^* \notin \{m^{(h)}\}_{h=1}^{q_s} \vee \psi_{i^*}^* \notin \{\psi_i^{(h)}\}_{h=1}^{q_s}$ holds, where $\psi_{i^*}^*$ is extracted from ψ_n^* as a signer structure from 1st signer to the target signer and $\psi_{i^*}^*$ includes exactly one honest signer; ψ_n^* includes ID^*; For all ID_j, ID_j in ψ_n^* is included in $\mathcal{L}$ except for ID^*; Each ID_j does not appear more than once in ψ_n^*. If all conditions hold, then $\mathcal{C}$ outputs *accept*. Otherwise, $\mathcal{C}$ outputs *reject*.

Note: We should discuss $(m^*, \psi_i^*) \notin \{(m^{(h)}, \psi_i^{(h)})\}_{h=1}^{q_s}$ as a natural security requirement. However, to the best of our knowledge, 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 $m^* \notin \{(m^{(h)}\}_{h=1}^{q_s} \vee \psi_i^* \notin \{\psi_i^{(h)})\}_{h=1}^{q_s}$. Even such a moderate model is not discussed in the standard model signature schemes [1, 11]. Through a discussion under this model, we prove that a proposed scheme guarantees that the validity of messages signed by an honest signer and his/her positions in the signing order.

We do not also consider *switching* among malicious signers. Suppose malicious signers ID_1 and ID_3 colludes against an honest singer ID_2. ID_1 and ID_3 may be able to compute some signature σ on m in $\psi = (ID_1 \parallel ID_2) \parallel ID_3$ after obtaining σ' on m in other signing order $\psi' = (ID_3 \parallel ID_2) \parallel ID_1$. To the best of our knowledge, there is no DLP-based scheme preventing such a forgery of the signing order and constructing such a scheme remains an open problem.

Definition 4. We say that an adversary $\mathcal{A}$ breaks an ordered multisignature scheme with $(t, q_c, q_s, l, n, \epsilon)$ if and only if a challenger $\mathcal{C}$ outputs *accept*, in the security game described above, with the probability greater than ϵ within the execution time t. Here, $\mathcal{A}$ can generate at most q_c certification queries and at most q_s signing queries, l is an upper bound on the length of the message output by $\mathcal{A}$, and n is an upper bound on the number of signers included in the forgery.

3.4 Our Approach

We construct a signature equation of an ordered multisignature scheme as $S = \prod_{i=1}^n g^{\alpha_i} \times \left(u' \prod_{j=1}^l u_j^{m_j}\right)^r (R)^{\sum_{i=1}^n it_i + v_i}$. As a main modification, we add new secret keys t_i and v_i, which are used by Boldyreva et al. [3], with an index representing signer's position i and the random number R as the third term, while the main formation in the scheme by Lu et al. [11] is kept. Hence, by classifying the following cases, the security of both the message and the signing order in our scheme can be proven: (case 1) $m^* \notin \{m^{(h)}\}_{h=1}^{q_s}$; (case 2) $m^* \in \{m^{(h)}\}_{h=1}^{q_s} \wedge \psi_i^* \notin \{\psi_i^{(h)}\}_{h=1}^{q_s}$. As described in section 4.2, $(u' \prod_{j=1}^l u_j^{m_j})^r (R)^{\sum_{i=1}^n it_i + v_i}$

can be written as $(g^{F(m)})^r(g^r)^{\sum_{i=1}^{n}(it_i+v_i)}$ by embedding polynomials $F(m)$. Here, we can embed a challenge of the CDH problem either in $(u'\prod_{j=1}^{l}u_j^{m_j})$ (case 1: the technique by Lu et al. [11]) or $T_i^iV_i$ (case 2: the technique by Boldyreva et al.), where T_i and V_i are public keys corresponding to t_i and v_i. Particularly, for case 2, we set a secret key v_i such that $T_i^iV_i = g^{v_i}$ for any i, and by using this setting we can remove the random number from the $\mathcal{A}$'s output when $\mathcal{A}$ forges the signing order. In paper [16], we set a secret key (t_i, v_i) such that $T_i^iV_i = 1$ for some i, and hence the simulation is stopped with non-negligible probability and the adversary can detect a difference with the original game. In addition, one can also use the technique by Lu et al. which breaks the Waters signature scheme when $\mathcal{A}$ forges the message (case 1).

4 Proposed Scheme

We assume that a trusted center to generate a public parameter exists. A message m in this scheme will be dealt as a bit-string of the form $\{0,1\}^l$ for all l.

4.1 Construction of the Scheme

The scheme consists of the following algorithms.

Setup: The trusted center generates a pairing parameter $(p, \mathbb{G}, \mathbb{G}_T, \boldsymbol{e})$ described in section 2.2. Then, the center generates random generators $g \in \mathbb{G}$ and $l+1$ generators $u', u_1, \cdots, u_l \in \mathbb{G}$. Finally, the center publishes the parameters $(p, \mathbb{G}, \mathbb{G}_T, \boldsymbol{e}, g, u', u_1, \cdots, u_l)$ as public parameter.

Key Generation: Given $(p, \mathbb{G}, \mathbb{G}_T, \boldsymbol{e}, g, u', u_1, \cdots, u_l)$, a signer ID_i chooses random numbers $\alpha_i, t_i, v_i \leftarrow \mathbb{Z}_p^*$, and sets $A_i = \boldsymbol{e}(g,g)^{\alpha_i}, T_i = g^{t_i}$ and $V_i = g^{v_i}$. His/Her secret key sk_i is (α_i, t_i, v_i) and the public key pk_i is (A_i, T_i, V_i).

Signing: Given m, a multisignature σ' by the previous signer and ψ_{i-1}, ID_i parses m as a bit-string $(m_1, \cdots, m_l) \in \{0,1\}^l$ and the signature σ' as (S', R'). First, ID_i verifies that the received signature σ' is a valid signature on m in ψ_{i-1} by using verification algorithm for $n = i'$ where $i' = i - 1$. If the verification algorithm outputs *reject*, ID_i aborts the process. If ID_i is the first signer in the signing group, then he/she sets $(S', R') = (1, 1)$ and $\prod_{j=1}^{i-1}T_j^jV_j = 1$, and skips the verification step described above. Otherwise, ID_i generates a random number $r_i \leftarrow \mathbb{Z}_p^*$ and computes $S = S' \cdot g^{\alpha_i}\left(u'\prod_{j=1}^{l}u_j^{m_j}\right)^{r_i}R^{it_i+v_i}\left(\prod_{j=1}^{i-1}T_j^jV_j\right)^{r_i}$ and $R = R' \cdot g^{r_i}$. Finally, ID_i sets $\psi_i = \psi_{i-1} \parallel ID_i$, then sends m, $\sigma = (S, R)$ and ψ_i to the next signer.

Verification Given m, σ, ψ_n, and $\{pk_i\}_{i=1}^n$ a verifier V parses m as a bit-string $(m_1, \cdots, m_l) \in \{0,1\}^l$ and σ as (S, R). V extracts each signer's public key (A_i, T_i, V_i) from $\{pk_i\}_{i=1}^n$ and verifies that $\frac{\boldsymbol{e}(S,g)}{\boldsymbol{e}\left(R, u'\prod_{j=1}^{l}u_j^{m_j}\right)\cdot\boldsymbol{e}\left(R, \prod_{i=1}^{n}T_i^iV_i\right)} \stackrel{?}{=} \prod_{i=1}^{n}A_i$ holds. If not, V outputs *reject*. Otherwise, V outputs *accept*.

4.2 Security Analysis

We prove the following theorem holds. The proof is similar as the Theorem 1 in [11] and the Theorem 3.3 in [3].

Theorem 5. The proposed ordered multisignature scheme is $(t, q_c, q_s, l, n, \epsilon)$-secure if and only if $(t_{CDH}, \epsilon_{CDH})$-CDH assumption holds, where $\epsilon_{CDH} = \frac{\epsilon}{16(l+1)q_s+\epsilon(q_s-1)}$ and $t_{CDH} = t + \mathcal{O}(q_c) + \mathcal{O}(nq_s) + \Psi$. Here, Ψ is depending on the number of signers n and the length of the message l.

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 by using the challenge in the problem and then generates a challenge in the proposed scheme from the Waters challenge. On the other hand, for case 2, he/she 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, (ϵ', t') for case 1 and (ϵ'', t'') for case 2. Then, we compute the whole probability ϵ_{CDH} and the whole computational time t_{CDH}.

In this proof, We assume that there exists exactly one signer whose secret key $\mathcal{A}$ does not know, and we call the signer a target ID^*. $\mathcal{B}$

has a certified-key list $\mathcal{L}$ and we denote by $x^{(j)}$ a value of j-th query for all x.

case 1: We construct an algorithm $\mathcal{B}$ which breaks the Waters signature scheme using $\mathcal{A}$. This step is almost same as the proof in [11].

Lemma 6. The proposed scheme is $(t, q_c, q_s, l, n, \epsilon)$-secure if and only 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 $t_W = t + \mathcal{O}(q_c) + \mathcal{O}(nq_s) + \Psi_1$, where Ψ_1 is the computational times for the final steps and is depending on the number of signers n.

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 $g, u', u_1, \cdots, u_l, p, \mathbb{G}, \mathbb{G}_T, \boldsymbol{e}$ 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 $T^* = g^{t*}, V^* = g^{v^*}, A^* = A_W$ as a target signer's public key. Here, let its corresponding secret key be α_W. Then, $\mathcal{B}$ runs $\mathcal{A}$ with $(p, \mathbb{G}, \mathbb{G}_T, \boldsymbol{e}, g, u', u_1, \cdots, u_l, A^*, T^*, V^*)$.

Certification Query: This step is almost same in [11].

Signing Query: This step is almost same in [11], but the following condition holds in the given query: Each ID_j does not appear more than once in $\psi_{i-1}^{(h)}$. Ffter obtaining a signature from $\mathcal{O}_W$, he/she computes $S = S' \times (R')^{\sum_{j=1}^{i} jt_j + v_j} \cdot g^{\sum_{j=1}^{i-1} \alpha_j}$ and sets $(S, R = R')$ as a ordered multisignature. This computation uses the re-randomization technique similarly with the paper [11].

Output: Also this step is almost same as the paper in [11], except that the following condition holds: Each ID_j does not appear more than once in ψ_n^*. After $\mathcal{A}$ outputs a forgery $\sigma^* = (S^*, R^*)$, $\mathcal{B}$ 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, $\mathcal{B}$ can output σ_W^* as the Waters signature scheme by setting $S_W^* = \frac{S^*}{g^{\sum_{j=1 \wedge j \neq i^*}^{n} \alpha_j} (R)^{\sum_{j=1}^{n} (jt_j + v_j)}}$ and $R_W^* = R^*$.

Finally, we evaluate the success probability ϵ_W and an execution time t_W of $\mathcal{B}$. Intuitively, this proof method is almost same as the method in [11] and there is no new event in which $\mathcal{B}$ aborts the simulation. Therefore, $\epsilon_W = \epsilon$ and $q_{W_s} = q_s$ hold. Similarly, the execution time is $t_W = t + \mathcal{O}(q_c) + \mathcal{O}(nq_s) + \Psi_1$, where Ψ_1 is the computational time for the final step and this value depends on the number of signers n. □

Here, We note following theorem [15].

Theorem 7. The Waters signature scheme is (t, q, ϵ)-secure if and only if (t', ϵ')-CDH assumption holds, where $\epsilon' = \frac{\epsilon}{16(l+1)q}$ and $t' = t$.

Proof (Sketch). The proof is given in [15]. □

This theorem implies that, when the proposed scheme is broken, we can construct an algorithm to solve the CDH problem with (ϵ', t'), where $\epsilon' = \frac{\epsilon}{16(l+1)q_s}$ and $t' = t + \mathcal{O}(q_c) + \mathcal{O}(nq_s) + \Psi_1$.

case 2: Next, we prove the security for case 2. This proof is based on the proof in [3].

Lemma 8. The proposed ordered multisignature scheme is $(t, q_c, q_s, l, n, \epsilon)$-secure if and only if (t'', ϵ'')-CDH assumption holds, where $\epsilon'' = \frac{\epsilon}{e(q_s - 1)}$ and $t'' = t + \mathcal{O}(q_c) + \mathcal{O}(nq_s) + \Psi_2$. Here, Ψ_2 is the computational times for the final steps, and is depending on the number of signers n and the length of the message l.

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-length vectors $\boldsymbol{x_i} \leftarrow \mathbb{Z}_p^l$ and $x' \leftarrow \mathbb{Z}_p$. For a message m, we define polynomials $F(m) = x' + \sum_{i=1}^{l} x_i m_i$, 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. $(u' \prod_{j=1}^{l} u_j^{m_j}) = g^{F(m)}$. 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 $A^* = \boldsymbol{e}(g^a, g^b)$. This means that $\mathcal{B}$ sets implicitly values including ab as the target signer's secret key. Then, $\mathcal{B}$ runs $\mathcal{A}$ with $(p, \mathbb{G}, \mathbb{G}_T, \boldsymbol{e}, g, u', u_1, \cdots, u_l, A^*, T^*, V^*)$.

Certification Query: This step is almost same in [3].

Signing Query: This step is almost same in [3], but the following conditions holds: Each ID_j does not appear more than once in $\psi_{i-1}^{(h)}$. He/She generates a random number $r \leftarrow \mathbb{Z}_p$ and computes $S = (g^b)^{-\frac{v^*}{t(i-k^*)}} ((g^a)^{it^*} (g^a)^{-t^* k^*} \times g^{v^*})^r \cdot (R')^{F(m^{(h)}) + \sum_{j=1}^{i-1} (jt_j + v_j)} \; g^{\sum_{j=1}^{i-1} \alpha_j}$ and $R = g^r (g^b)^{-\frac{1}{t^*(i-k^*)}}$ as a ordered multisignature. This computation uses the re-randomization technique similarly in [3].

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^*. This step is almost same in [3], but the following condition holds for this forgery; Each ID_j does not appear more than once in ψ_i^*, where ψ_i^* is extracted from ψ_n^*. If $i^* = k^*$, $\mathcal{B}$ can solve the CDH problem since the forgery can be written as $S^* = g^{ab} \prod_{i=1 \wedge i \neq i^*}^{n} g^{\alpha_i} \ (g^X)^r$ and $R^* = g^r$, where $X = F(m) + \sum_{j=1 \wedge j \neq k^*}^{n} (jt_j + v_j) + v^*$. Then, $\mathcal{B}$ computes $g^{ab} = \frac{S^*}{(R^*)^X \prod_{j=1 \wedge j \neq k^*}^{n} g^{\alpha_j}}$.

Let $i^{(h)}$ be the position of the target signer for h-th query, and then 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 [3]. Then, $\epsilon'' \geq \epsilon \cdot \frac{1}{e(q_s - 1)}$ holds. The $\mathcal{B}$'s execution time t' is $t'' = t + \mathcal{O}(q_c) + \mathcal{O}(nq_s) + \Psi_2$, where Ψ_2 means the computational time for the final step and this value is depending on the number of signers n and the length of the message l. □

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 the lemma 6, the theorem 7 and the lemma 8, $\epsilon_{CDH} = \alpha \cdot \beta \cdot \frac{\epsilon}{16(l+1)q_s} + (1 - \alpha) \cdot (1 - \beta) \frac{e}{q_s - 1} \cdot \epsilon$ holds. In order to be a complete proof, we analysis values of α and β. Let $f(\alpha, \beta)$ denote $\alpha \cdot \beta \cdot \frac{\epsilon}{a} + (1 - \alpha) \cdot (1 - \beta) \frac{1}{e(q_s - 1)} \cdot \epsilon$, where $a = 16(l+1)q_s$ and $b = e(q_s - 1)$ as constants shortly. and 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 for t' and t''. Therefore, $t_{CDH} = \max\{t', t''\} = t'' = t + \mathcal{O}(q_c) + \mathcal{O}(nq_s) + \Psi_2$ holds. □

5 Evaluation of the Scheme

We compare the performance of the proposed scheme with simple ordered multisignature schemes, given by aggregate signature schemes, in the standard model [1, 11] with respect to the signing cost, the verification cost, the signature size, the public key size and the type of the scheme. Here, we note that a multiplication between the pairing computation, i.e. a multiplication over $\mathbb{G}_T$, is higher than a multiplication over $\mathbb{G}$.

As shown table. 1, in compared with [1, 11], the signature size of the proposed scheme is the same as them, and the number of the pairing computation in the verification cost and the size of the public keys are independent of the length of the message. Although our proposed scheme has the linear computational cost for the multiplication of the pairing with respect to the number of signers, generally speaking the number of signers is much fewer than the number of bits in the message. For instance, according to Kanaoka et al. [9] a distance between two routers on the Internet can be coverall for 20 hops. In other words, we consider that in S-BGP the number of signers, that are routers, is at most 20. This value is obviously smaller than the parameter size.

Therefore, our proposed scheme is the most practical for an implementation on several devices such as router which processes many packets with a small amount of memory.

6 Conclusion

In this paper, we proposed an ordered multisignature scheme which is a signature scheme verifying both the validity of the message and the signing order. Most of the existing ordered multisignature schemes adopt the random oracle model to analyze the security. We also pointed out a bug in [3] and revised the security model toward being moderate. To the best of our knowledge, our scheme is the first CDH-based scheme achieving all the following conditions: the rigorous security analysis in the standard model under the moderate attack model, the fixed number of pairing computation and the fixed size public key with respect to the length of the message. Here, we note that the security without the switching in the standard model is proven, and proving the security with the switching described in section 3.3 is still an open problem. Therefore, we consider that we will discuss an ordered multisignature scheme secure against the switching and also a scheme not requiring the moderate attack model in a future. We also plan to extend the our proposed scheme for aggregate signature scheme and to implement the data-plane security with the proposed scheme.

Table 1: Evaluation of the Proposed Scheme

Related Work	Signing Cost for i-th Signer	Verification Time for n Signers	Signature Size	Public Key	Type of Scheme
Ahn et al.[1]	$E(1+l)+E(3)$ $+E(2)+E(1)$	$(3+l)\mathcal{P}+(3+l)\mathcal{MP}$ $+2E(n)+E(3)+n\mathcal{R}$	$2L(p)$	$L(p)$	Simple Ordered
Lu et al.[11]	$E((i+1)l+1)$ $+5+l\mathcal{R}$	$2\mathcal{P}+n\mathcal{MP}$ $nl+\mathcal{I}$	$2L(p)$	$(l+2)L(p)$	Simple Ordered
Our Scheme	$E((i+1)l+2)$ $+5+l\mathcal{R}+(2i)$	$3\mathcal{P}+(1+n)\mathcal{MP}$ $+E(2n)+E(n+1)+\mathcal{I}$	$2L(p)$	$3L(p)$	Ordered

We denote by l the message length, by $\mathcal{P}$ the computational cost of bilinear map, by $\mathcal{MP}$ the multiplication cost between bilinear map, i.e. the multiplication over $\mathbb{G}_T$, and by $E(n) := (\frac{n}{2}+1)L(p)-1$ the required number of modulo-p multiplication for computing $g_1^{a_1}\cdots g_n^{a_n}$ with $g_i \in \mathbb{Z}_p^*$ and $a_i \in \mathbb{Z}_q$, where $L(p)$ denotes the binary length of p. We denote by $\mathcal{R}$ the ratio of the computational cost of multiplication in $\mathbb{Z}_p^*$ to that of multiplication modulo p in $\mathbb{F}_p$ and by $\mathcal{I}$ the computational cost for inversion in $\mathbb{F}_p$. For the type of the scheme, Simple Ordered means simple ordered multisignature scheme and Ordered means truly ordered multisignature scheme which is not simple one.

Acknowledgements

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.

Reference

[1] Ahn, J. H., Green, M. and Hohenberger, S. (2010) Synchronized Aggregate Signatures: New Definitions, Constructions and Applications. *Cryptology ePrint Archive*, pp.1-26, `http://eprint.iacr.org/2010/422`.

[2] Bellare, M. and Rogaway, M. (1993) Random Oracle are Practical: A Paradigm for Designing Efficient Protocols. *Proc. of CCS 1993*, USA, pp.62-73, ACM Press.

[3] Boldyreva, A., Gentry, C., O'Neill, A. and Yum, D. H. (2007) Ordered Multisignatures and Identity-Based Sequential Aggregate Signatures, with Applications to Secure Routing. *Proc. of CCS 2007*, USA, pp.276-285, ACM Press. Also in available in `http://www.cc.gatech.edu/~aboldyre/papers/bgoy.pdf`.

[4] Boneh, D., Gentry, C., Lynn, B. and Shacham, H. (2003) Aggregate and Verifiability Encrypted Signatures from Bilinear Maps. *Proc. of EUROCRYPT 2003*, Poland, pp.416-432, LNCS 2656.

[5] Canetti, R., Goldreich, O. and Halevi, S. (2004) The Random Oracle Methodology, Revisited. *JACM*, Vol.51, No.4, pp.557-594.

[6] Doi, H., Mambo, M. and Okamoto, E. (1998) Multisignature Schemes Using Structured Group ID. *TECHNICAL REPORT OF IEICE*, ISEC98-53, pp.43-48, IEICE.

[7] Hou, W. (2010) An Ordered Multisignature without Random Oracles. *Proc. of CMC 2010*, China, pp.21-25, IEEE.

[8] Itakura, K. and Nakamura, K. (1983) A Public-key Cryptosystem Suitable for Digital Multi-signatures. *TIPSJ*, Vol.24, No.4, pp.474-480.

[9] Kanaoka, A., Masayuki, O., Katsuno, Y., and Okamoto, E. (2011) Probabilistic Packet Marking Method Considering Topology Property for Efficiency Re-building DoS Attack Paths. *TIPSJ*, Vol.52, No.3, pp.929-939.

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

[11] Lu, S., Ostrovsky, R., Sahai, A., Shacham, H. and Waters, B. (2006) Sequential Aggregate Signatures and Multisignatures Without Random Oracle. *Proc. of EUROCRYPT 2006*, Russia, pp.465-485, LNCS 4004.

[12] Rekhter, Y., and Li, T. (1995) A Border Gateway Protocol 4 (BGP-4). *RFC 1771*, `http://www.ietf.org/rfc1771.txt`.

[13] Rückert, M., and Schröder, D. (2009), Aggregate and Verifiability Encrypted Signatures from Multilinear Maps without Random Oracles. *Proc. of ISA 2009*, Korea, pp.750-759, LNCS 5576.

[14] Schröder, D. (2011) How to Aggregate the CL Signature Scheme. *Proc. of ESORICS 2011*, Belgium, pp.298-314, LNCS 6879.

[15] Waters, B. (2005) Efficient identity-based encryption without random oracles. *Proc. of EUROCRYPT 2005*, Denmark, pp.114-127, LNCS 3494.

[16] Yanai, N., Chida, E., and Mambo, M. (2011) A Secure Ordered Multisignature Scheme without Random Oracles, *Proc. of SCIS 2011*, Japan, 3A1-2.