PAPER *Special Section on Information and Communication System Security*

Multi-Group Signature Scheme for Simultaneous Verification by Neighbor Services

Kenta NOMURA[†*a)], ***Nonmember***, **Masami MOHRI**[††], **Yoshiaki SHIRAISHI**[†], ***and*** **Masakatu MORII**[†], ***Senior Members***

SUMMARY We focus on the construction of the digital signature scheme for local broadcast, which allows the devices with limited resources to securely transmit broadcast message. A multi-group authentication scheme that enables a node to authenticate its membership in multi verifiers by the sum of the secret keys has been proposed for limited resources. This paper presents a transformation which converts a multi-group authentication into a multi-group signature scheme. We show that the multi-group signature scheme converted by our transformation is existentially unforgeable against chosen message attacks (EUF-CMA secure) in the random oracle model if the multi-group authentication scheme is secure against impersonation under passive attacks (IMP-PA secure). In the multi-group signature scheme, a sender can sign a message by the secret keys which multiple certification authorities issue and the signature can validate the authenticity and integrity of the message to multiple verifiers. As a specific configuration example, we show the example in which the multi-group signature scheme by converting an error correcting code-based multi-group authentication scheme.

key words: *internet of things, local broadcast, digital signature, fiat-shamir transform, low energy*

1. Introduction

Internet of Things (IoT) represents all things connect to the Internet and share information with each other. Such information allows devices to mutually control and operate. The source and destination devices should mutually authenticate each other. In each independent application and services, devices should be authenticated by independent secret information. In asymmetric key-based authentication, those to be authenticated, i.e. a prover, has a pair of a public/secret key and those to authenticate, i.e. a verifier, has the public key. The prover demonstrates to the verifier that it is indeed in possession of the secret key of corresponding to the public key via messaging protocol. Application systems which behave as the verifier generally have their own public key infrastructure (PKI). That is to say the prover uses a different secret key certified by its authority for each application to perform authentication.

In such an environment, the amount of transmission data required to authenticate membership proportionally increase with increasing the number of applications and services. A malicious user may be able to successfully masquerade as a valid prover due to the leak of a single secret key since the verifier identifies the prover by a single secret key for each application. For such a security issue, the multi-group authentication scheme proposed by Halford [1], which proves by the sum of the secret keys. It suppresses the increase in the amount of transmission data compared to the naïve approach. In addition, it is impossible to masquerade for a malicious user unless all the secret keys are revealed because the verifier authenticates by the sum of a plurality of secret keys.

Some physical devices are controlled by the information stored and exchanged in the network, thus any accidental or malicious alteration of the information may cause serious trouble. Therefore, a sender should attach a digital signature to the transmission data in order to safely use the IoT technology. The digital signature which validates the authenticity and integrity of the data is signed by different key pairs for each application or service like authentication. In that case, the sender is required to make an individual signature for the same data to different destination. We focus a multi-group signature which a message is signed by multiple keys the independent PKIs issue like the multi-group authentication.

In this paper, according as Fiat-Shamir paradigm [2], we describe how to construct multi-group signature schemes from multi-group authentication schemes and give a security proof. We show the concrete multi-group signature scheme where digital signature is generated by multiple secret keys so as to verify multiple verifiers simultaneously.

2. Authentication Scheme

2.1 Classification of Schemes

We classify the authentication scheme into three approaches, knowledge-based authentication (e.g., ID/password) [3]–[6], key-based authentication (e.g., public/secret key) [7]–[9], and the authentication based on an interactive protocol system which involves a prover and verifier. Zero-knowledge proof based protocols seem to be suitable for WSN to reduce the energy consumption [10]. As one of the authentication based on zero-knowledge proof, Stern has proposed code-based authentication [11]. More efficient schemes based on the Stern's scheme have proposed [12]–[15].

Manuscript received September 8, 2016.
Manuscript revised January 30, 2017.
Manuscript publicized May 18, 2017.

†The authors are with Kobe University, Kobe-shi, 657–8501 Japan.
††The author is with Gifu University, Gifu-shi, 501–1193 Japan.
*Presently, with PwC Cyber Services LLC.
a) E-mail: kenta.nomura@pwc.com

DOI: 10.1587/transinf.2016ICP0029

Copyright © 2017 The Institute of Electronics, Information and Communication Engineers

The simplest authentication schemes verify the validity of the sender in the two party model such that a verifier authenticate a prover. Since the use cases have expanded and the application has diversified, the authentication scheme for multiple provers or verifiers have proposed. As the example of some provers, authentication server verifies identity of multiple requests of users at the same time to decrease the load of the system in [16] and the authentication schemes which can determine whether all provers participated in a group communication belong to the same group has proposed [17], [18]. As the example of some verifiers, a multi-group authentication scheme which authenticates provers by the sum of the secret keys has proposed [1].

2.2 Entities

There are two entities in the authentication scheme; prover and verifier. We focus the authentication scheme through a three-pass interaction between the prover and the verifier.

prover: A prover, holding a secret key, sends a message called a *commitment* to the verifier and provides a *response* following a *challenge*.

verifier: A verifier receives *commitments* and returns a *challenge* consisting of a random string of some length. After the verifier receives a *response* from the prover, the verifier calculates the estimated *response* value by a public key, the *commitment* and the *challenge*. If the estimated value corresponds to the *response* obtained from the prover, the verifier authenticates the prover.

2.3 Authentication Scheme Using a Single Key

2.3.1 Algorithms

The authentication scheme $ID = (K, Co, Ch, R, V_A)$ consists of the five algorithms where K is the key generation algorithm, taking as input a security parameter 1^k and returning a public key and secret key pair (pk, sk); Co is the *commitment* algorithm, taking as input sk and returning a *commitment Cmt*; Ch is the *challenge* algorithm, taking as input the length c of the verifier's challenge and returning a c bits *challenge*; R is the *response* algorithm, taking as input (sk, Ch) and returning a *response Rsp*; V_A is the verification algorithm which verifiers verify provers, taking as input (pk, Cmt, Ch, Rsp) and comparing Rsp and a *response* value obtained from (pk, Cmt, Ch). That is, V_A returns 1 as decision Dec if and only if both coincide.

2.3.2 Security Definition

Secure schemes prevent a malicious adversary impersonating the prover without the knowledge of the valid secret key. The definition that an authentication scheme is *secure against impersonation under passive attacks* (IMP-PA secure) [19] is shown as follows:

Definition 1 [IMP-PA security of authentication schemes] Let $ID = (K, Co, Ch, R, V_A)$ be an authentication scheme, and let I be an impersonator, be st its state, and be k the security parameter. Define the advantage of I as $\mathbf{Adv}^{\text{ima-pa}}_{ID,I}(k) = \Pr\left[\mathbf{Exp}^{\text{ima-pa}}_{ID,I}(k) = 1\right]$ where the experiment $\mathbf{Exp}^{\text{ima-pa}}_{ID,I}(k)$ in the equation is

Experiment $\mathbf{Exp}^{\text{ima-pa}}_{ID,I}(k)$

$$(pk, sk) \xleftarrow{\$} K(k);\ st \,\|\, Cmt \xleftarrow{\$} I^{\mathrm{Tr}^{ID}_{pk,sk,k}}(pk)$$
$$Ch \xleftarrow{\$} \{0,1\}^{c(k)};\ Rsp \xleftarrow{\$} I(st, Ch)$$
$$\mathrm{Dec} \leftarrow V_A(pk, Cmt \,\|\, Ch \,\|\, Rsp);\ \text{return Dec}$$

Then, we associate to an ID and each (pk, sk) a randomized *transcript generation oracle* which takes no inputs and returns a random transcript of an "honest" execution, namely:

Function $\mathrm{Tr}^{ID}_{pk,sk,k}$

$$R_p \xleftarrow{\$} \mathrm{Coins}_P(k)$$
$$Cmt \leftarrow Co(sk; R_p);\ Ch \xleftarrow{\$} \{0,1\}^{c(k)}$$
$$Rsp \leftarrow R(sk, Cmt \,\|\, Ch; R_P)$$
$$\text{return } Cmt \,\|\, Ch \,\|\, Rsp$$

We say that an ID is *secure against impersonation under passive attacks* if the $\mathbf{Adv}^{\text{ima-pa}}_{ID,I}(k)$ is negligible for every impersonator I of probabilistic polynomial in the security parameter k.

2.4 Multi-Group Authentication

When different authentication services use a different public key pair, namely provers possess multiple public key pairs, the authentication scheme of Sect. 2.3 must perform the authentication protocol as much as the number of key pairs in order to simultaneously receive multiple authentication services. The multi-group authentication scheme [1] which enables a prover to simultaneously authenticate its membership is proposed. In that scheme, a prover demonstrates to multiple verifiers that it is indeed in possession of the multiple secret keys to be authenticated by multiple verifiers.

2.4.1 Algorithm

The multi-group authentication scheme $mg-ID = (mgK, mgCo, Ch, mgR, mgV_A)$ consists of the five algorithms where mgK is the key generation algorithm, taking as input a security parameter 1^k and the number M of keys required and returning M public key and secret key pairs $\{(pk_i, sk_i)\}^M_{i=1}$; $mgCo$ is the *commitment* algorithm, taking as input N secret keys $\{sk_i\}^N_{i=1}$ to demonstrate its knowledge of N secret keys $\{sk_i\}^N_{i=1}$ and returning a *commitment Cmt*; Ch is the same in Sect. 2.3.1; mgR is the *response* algorithm, taking as input $(\{sk_i\}^N_{i=1}, Ch)$, where $\{sk_i\}^N_{i=1}$ are same keys in input of $mgCo$, and returning a *response Rsp*; mgV_A is the verification algorithm which verifiers verify provers, taking as input $(\{pk_i\}^N_{i=1}, Cmt, Ch, Rsp)$, where $\{pk_i\}^N_{i=1}$ are corresponding to $\{sk_i\}^N_{i=1}$ in input of $mgCo$ and mgR, and comparing Rsp and a *response* value obtained from $(\{pk_i\}^N_{i=1}, Cmt, Ch)$.

mgV_A returns 1 as decision Dec if and only if Rsp and the *response* value are coincided. Provers run $mgCo$ and mgR. Verifiers run Ch and mgV_A.

2.4.2 Security Definition

We define IMP-PA secure of the multi-group authentication scheme. In Definition 1, an impersonator I tries to impersonate the prover who has a single secret key. However, an impersonator I tries to impersonate the prover who has N secret keys in following Definition 2. Therefore, we assume that I can get outputs of a *transcript generation oracle* where a prover has N secret keys.

Definition 2 [IMP-PA security of multi-group authentication schemes] Let $mg-ID = (mgK, mgCo, Ch, mgR, mgV_A)$ be a multi-group authentication scheme, and let I be an impersonator, be st its state, and be k the security parameter. Define the advantage of I as $\mathbf{Adv}^{\text{ima-pa}}_{mg-ID,I,N}(k) = \Pr\left[\mathbf{Exp}^{\text{ima-pa}}_{mg-ID,I,N}(k) = 1\right]$ where the experiment $\mathbf{Exp}^{\text{ima-pa}}_{mg-ID,I,N}(k)$ in the equation is

Experiment $\mathbf{Exp}^{\text{ima-pa}}_{mg-ID,I,N}(k)$

$$\{(pk_i, sk_i)\}_{i=1}^{M} \stackrel{\$}{\leftarrow} K(k)$$
$$st \,\|\, Cmt \stackrel{\$}{\leftarrow} I^{\mathrm{Tr}^{mg-ID}_{N,pk,sk,k}}(\{pk_i\}_{i=1}^{M})$$
$$Ch \stackrel{\$}{\leftarrow} \{0,1\}^{c(k)};\ Rsp \stackrel{\$}{\leftarrow} I(st, Ch)$$
$$\mathrm{Dec} \leftarrow mgV_A(\{pk_i\}_{i=1}^{M}, Cmt \,\|\, Ch \,\|\, Rsp)$$
return Dec

Then, we associate to a $mg-ID$ and N key pairs $\{(pk_i, sk_i)\}_{i=1}^{N}$ a randomized *transcript generation oracle* which takes no inputs and return a random transcript of an "honest" execution, namely:

Function $\mathrm{Tr}^{mg-ID}_{N,pk,sk,k}$

$$R_p \stackrel{\$}{\leftarrow} \mathrm{Coins}_P(k)$$
$$Cmt \leftarrow Co(\{sk_i\}_{i=1}^{M}; R_p);\ Ch \stackrel{\$}{\leftarrow} \{0,1\}^{c(k)}$$
$$Rsp \leftarrow R(\{sk_i\}_{i=1}^{M}, Cmt \,\|\, Ch; R_P)$$
return $Cmt \,\|\, Ch \,\|\, Rsp$

We say that a $mg-ID$ is *secure against impersonation under passive attacks* if the $\mathbf{Adv}^{\text{ima-pa}}_{mg-ID,I,N}(k)$ is negligible for every impersonator I of probabilistic polynomial in the security parameter k.

3. Signature Scheme

3.1 Entities

There are two entities in signature schemes; signer and verifier.

signer: A signer generates a signature to sign a message by using a secret key.

verifier: A verifier verifies the validity of signature for a message by using a public key.

3.2 Signature Scheme Using a Single Key

3.2.1 Algorithms

The signature scheme $DS = (K, S, V_s)$ consists of the three algorithm where K is the key generation algorithms, taking as input a security parameter 1^k and returning a public key and secret key pair (pk, sk); S is the signing algorithm, taking as input sk and a message m and returning a signature σ; V_s is the verification algorithm, taking as input (pk, m, σ) and checking whether σ is a valid signature for m. That is, V_S returns 1 as decision Dec if and only if it is valid. The signing algorithm may be randomized, drawing coins from a space $\mathrm{Coins}_S(k)$, but the verification algorithm is deterministic.

3.2.2 Security Definition

Security of a signature scheme is defined as kinds of attacks and difficulty in forgery. We describe that a signature scheme is existentially unforgeable against adaptive chosen-message attacks (EUF-CMA secure) [20] in the random oracle model [21]. The adversary F, called a forger, gets the usual signing oracle plus direct access to the random oracle and wins if it outputs a valid signature of a new message. We let $[\{0,1\}^* \to \{0,1\}^c]$ denote the set of all maps from $\{0,1\}^*$ to $\{0,1\}^c$. The notation $h \stackrel{\$}{\leftarrow} [\{0,1\}^* \to [0,1]^c]$ is used to mean that we select a hash function h as random from this set.

Definition 3 [EUF-CMA security of signature schemes] Let $DS = (K, S, V_s)$ be a digital signature scheme, let F be a forger and k the security parameter. Define the advantage of F as

$$\mathbf{Adv}^{\text{euf-cma}}_{DS,F}(k) = \Pr\left[\mathbf{Exp}^{\text{euf-cma}}_{DS,F}(k) = 1\right]$$

where the experiment $\mathbf{Exp}^{\text{euf-cma}}_{DS,F}(k)$ in the equation is

Experiment $\mathbf{Exp}^{\text{euf-cma}}_{DS,F}(k)$

$$h \stackrel{\$}{\leftarrow} [\{0,1\}^* \to [0,1]^c]$$
$$(pk, sk) \stackrel{\$}{\leftarrow} K(k);\ (m, \sigma) \stackrel{\$}{\leftarrow} F^{S^h_{sk}(\cdot), h(\cdot)}(pk)$$
$$\mathrm{Dec} \leftarrow V^h_s(pk, m, \sigma)$$
If m was previously queried to $S^h_{sk}(\cdot)$

Then return 0 Else return Dec

We say that a DS is existentially unforgeable against adaptive chosen-message attacks (EUF-CMA secure) if $\mathbf{Adv}^{\text{euf-cma}}_{DS,F}(k)$ is negligible every forger F of probabilistic polynomial in the security parameter k.

3.3 Multi-Group Signature Scheme

3.3.1 Algorithms

In the signature scheme of Sect 3.2, a signer makes signature by using a single key. In contrast, in the multi-group signature scheme, the signer makes signature by using multiple keys. The multi-group signature scheme $mg - DS = (mgK, mgS, mgV_s)$ consists of the three algorithms where mgK is the key generation algorithm, taking as input a security parameter 1^k and the number M of keys required and returning M public key and secret key pairs $\{(pk_i, sk_i)\}_{i=1}^{M}$; mgS is the signing algorithm, taking as input N secret keys $\{sk_i\}_{i=1}^{N}$ to demonstrate its knowledge of N keys and a message m and returning a signature σ; mgV_s is the verification algorithm, taking as input $(\{pk_i\}_{i=1}^{N}, m, \sigma)$, where $\{pk_i\}_{i=1}^{N}$ are corresponding to $\{sk_i\}_{i=1}^{N}$ in input of mgS, and checking whether σ is a valid signature for m. That is, mgV_S returns 1 as decision Dec if and only if it is valid. The signing algorithm may be randomized and the verification algorithm is deterministic like Sect. 3.2.1.

3.3.2 Security Definition

The definition that a multi-group signature scheme is EUF-CMA secure in the random oracle model is shown here. In Definition 3, a forger F forges a signature made by using a single secret key. However, the forger against multi-group signature schemes will forge a signature made by using multiple secret keys. Thus, F can get the usual signing oracle plus direct access to the random oracle for multiple secret keys.

Definition 4 [EUF-CMA security of a multi-group signature scheme]
Let $mg - DS = (mgK, mgS, mgV_s)$ be a multi-group signature scheme, let F be a forger and k the security parameter. Define the advantage of F as

$$\mathbf{Adv}^{\mathrm{euf-cma}}_{mg-DS,F,N}(k) = \Pr\left[\mathbf{Exp}^{\mathrm{euf-cma}}_{mg-DS,F,N}(k) = 1\right]$$

where the experiment $\mathbf{Exp}^{\mathrm{euf-cma}}_{mg-DS,F,N}(k)$ in the equation is
Experiment $\mathbf{Exp}^{\mathrm{euf-cma}}_{mg-DS,F,N}(k)$

$$h \xleftarrow{\$} [\{0,1\}^* \to [0,1]^c]$$
$$\{(pk_i, sk_i)\}_{i=1}^{M} \xleftarrow{\$} mgK(k; M)$$
$$(m, \sigma) \xleftarrow{\$} F^{S^h_{N,sk}(\cdot),h(\cdot)}(\{pk_i\}_{i=1}^{N})$$
$$\mathrm{Dec} \leftarrow mgV^h_s(\{pk_i\}_{i=1}^{N}, m, \sigma)$$
If m was previously queried to $S^h_{sk}(\cdot)$
Then return 0 Else return Dec

We say that a $mg - DS$ is existentially unforgeable against adaptive chosen-message attacks (EUF-CMA secure) if $\mathbf{Adv}^{\mathrm{euf-cma}}_{mg-DS,F,N}(k)$ is negligible every forger F of probabilistic polynomial in the security parameter k.

4. Transforming Into a Multi-Group Signature Scheme

The Fiat-Shamir (FS) transformation is a general method to construct signature schemes from authentication schemes. The security of signature constructed by FS transformation, namely FS-type signature, is discussed in several literature. Pointcheval *et al.* [22] showed that an FS-type signature is EUF-CMA secure in random oracle model if the underlying authentication scheme is honest-verifier zero-knowledge proof of knowledge. Abdalla *et al.* [23] relaxed the below condition. More precisely, they proved the equivalence between the EUF-CMA security of an FS-type signature and the IMP-PA security of the underlying authentication scheme in the random oracle model. This result indicates that the IMP-PA security of the underlying authentication schemes is essential for proving the security of FS-type signatures in the random oracle model. In this paper, we show the below result is satisfied even multi-group.

4.1 The Fiat-Shamir Transformation

A signer computes a commitment Cmt just as a prover would at signing a message m. In authentication schemes, a prover receives a challenge from a verifier. A signer hashes $Cmt \,\|\, m$ using a public hash function h to obtain a challenge $Ch = h(Cmt \,\|\, m)$, then computes a response Rsp just as a prover would, and sets the signature of m to $Cmt \,\|\, Rsp$. Let $ID = (K, Co, Ch, R, V_A)$ and s: N $\to$ N be an authentication scheme and a function which we call the seed length, respectively. The Fiat-Shamir transformation associates ID with a signature scheme $DS = (K, S^h, V^h_s)$. The signing and verifying algorithms are defined as follow:
Signing algorithm $S^h(sk, m)$

$$R \xleftarrow{\$} \{0,1\}^{s(k)};\ R_p \xleftarrow{\$} \mathrm{Coins}_P(k)$$
$$Cmt \leftarrow Co(sk; R_p)$$
$$Ch \leftarrow h(R \,\|\, Cmt \,\|\, m)$$
$$Rsp \leftarrow R(sk, Cmt \,\|\, Ch; R_p)$$
return $R \,\|\, Cmt \,\|\, Rsp$

Verifying algorithm $V^h_s(pk, m, \sigma)$

parse σ as $R \,\|\, Cmt \,\|\, Rsp$
$$Ch \leftarrow h(R \,\|\, Cmt \,\|\, m)$$
$$\mathrm{Dec} \leftarrow V_A(pk, Cmt \,\|\, Ch \,\|\, Rsp)$$
return Dec

DS has the same key generation algorithm as ID, and the output length of a hash function equals the challenge length of ID.

4.2 Transformation from a Multi-Group Authentication Scheme to a Multi-Group Signature Scheme

This section presents the transformation from a multi-group authentication scheme to a multi-group signature scheme satisfied the definition in Sect. 3.3.

Let $mg - ID = (mgK, mgCo, Ch, mgR, mgV_A)$ be a multi-group authentication scheme and let s: N $\to$ N be a function which we call the seed length. We associate to these a multi-group signature scheme $mg - DS =$

(mgK, mgS^h, mgV_s^h). It has the same key generation algorithm as the multi-group authentication scheme, and the output length of the hash function equals the challenge length of the authentication scheme. The signing and verifying algorithms are defined as follow:
Signing algorithm $mgS^h(\{sk_i\}_{i=1}^N, m)$

$$\begin{aligned}
&R \xleftarrow{\$} \{0,1\}^{s(k)};\ R_p \xleftarrow{\$} \mathrm{Coins}_P(k)\\
&Cmt \leftarrow mgCo(\{sk_i\}_{i=1}^N; R_p)\\
&Ch \leftarrow h(R \,\|\, Cmt \,\|\, m)\\
&Rsp \leftarrow mgR(\{sk_i\}_{i=1}^N, Cmt \,\|\, Ch; R_p)\\
&\text{return } R \,\|\, Cmt \,\|\, Rsp
\end{aligned}$$

Verifying algorithm $mgV_s^h(\{pk_i\}_{i=1}^N, m, \sigma)$

$$\begin{aligned}
&\text{parse } \sigma \text{ as } R \,\|\, Cmt \,\|\, Rsp\\
&Ch \leftarrow h(R \,\|\, Cmt \,\|\, m)\\
&\mathrm{Dec} \leftarrow mgV_A(\{pk_i\}_{i=1}^N, Cmt \,\|\, Ch \,\|\, Rsp)\\
&\text{return Dec}
\end{aligned}$$

Note that the signing algorithm is randomized, using a random type whose length is $s(k)$ plus the length of the random tape of the prover. Furthermore, the chosen random seed is include as part of the signature, to make verification possible. Thus, we construct the multi-group signature scheme $mg-DS = (mgK, mgS^h, mgV_s^h)$ from the multi-group authentication scheme.

4.3 Security Proof

We use the concept of min-entropy [24], which is also quoted in [23], to measure how likely it is for a commitment generated by the prover of an authentication to collide with a specific value.

Definition 5 [Min-Entropy of Commitments]
Let $mg-ID = (mgK, mgCo, Ch, mgR, mgV_A)$ be a multi-group authentication scheme. Let $k \in \mathrm{N}$, and $\{(pk_i, sk_i)\}_{i=1}^N$ be key pairs generated by mgK on input k. Let $(\{sk_i\}_{i=1}^N) = \{mgCo(\{sk_i\}_{i=1}^N; R_P) : R_p \in \mathrm{Coins}_P(k)\}$ be the set of commitments associated to N secret keys $\{sk_i\}^N$, where $N \leq M$. We define the maximum probability that a commitment takes on a particular value via

$$\alpha(\{sk_i\}_{i=1}^N) = \max_{Cmt \in C(\{sk_i\}_{i=1}^N)} \left\{ \Pr\left[\begin{pmatrix} mgCo(\{sk_i\}_{i=1}^N; R_P) \\ = Cmt\colon\ R_P \xleftarrow{\$} \mathrm{Coins}_P(k) \end{pmatrix} \right] \right\}.$$

Then, the min-entropy function associated to $mg-ID$ is defined as follows:

$$\beta(k) = \min_{\{sk_i\}_{i=1}^N} \left\{ \log_2 \frac{1}{\alpha(\{sk_i\}_{i=1}^N)} \right\},$$

where minimum is over all $\{(pk_i, sk_i)\}_{i=1}^N$ generated by mgK on input k.

It is proven that the Theorem 1 for security of the multi-group signature scheme as follows:

Theorem 1
Let $mg-ID = (mgK, mgCo, Ch, mgR, mgV_A)$ be a multi-group authentication scheme, let $s(\cdot)$ be a seed length, and let $mg-DS = (mgK, mgS^h, mgV_s^h)$ be the multi-group signature scheme as per Sect. 4.1. Let $\beta(\cdot)$ be the min-entropy function associated to a $mg-ID$. Let F be an adversary attacking a $mg-DS$ in the random oracle model, having time-complexity $t(\cdot)$, making $q_s(\cdot)$ sign-oracle queries and $q_h(\cdot)$ hash-oracle queries. Then there exists an impersonator I attacking a $mg-ID$ such that

$$\begin{aligned}
&\mathbf{Adv}^{\mathrm{euf-cma}}_{mg-DS,F,N}(k)\\
&\leq (1 + q_h(k)) \cdot \mathbf{Adv}^{\mathrm{ima-pa}}_{mg-ID,I,N}(k) + \frac{[1 + q_h(k) + q_s(k)] \cdot q_s(k)}{2^{s(k)+\beta(k)}}.
\end{aligned} \tag{1}$$

Furthermore, I has time-complexity $t(\cdot)$ and makes at most $q_s(\cdot)$ queries to its transcript oracle. ■

We will prove Theorem 1 by referring to [23] and using code-based game-playing [25] which is quoted in [23]. We let $G_i^A \Rightarrow y$ donate the game G_i outputs with an adversary A takes value y. In the code-based game-playing, we use the Fundamental Lemma [25] in order to determine the upper limit of the random variable. We can apply the Fundamental Lemma only when the game G_i and G_{i+1} meets an equivalence relation on games called *identical until* **bad**. We say that G_i and G_{i+1} are *identical until* **bad** if their code is the same until one is substituted for the flag **bad**.

Lemma 1 Let G_i, G_j be *identical until* **bad** games, and A be an adversary. Then,

$$\Pr\left[G_i^A \Rightarrow 1\right] - \Pr\left[G_j^A \Rightarrow 1\right] \leq \Pr[G_i \textit{ sets } \textbf{bad}].$$

Lemma 2 Let G_i, G_j be *identical until* **bad** games, and A be an adversary. We let Good_i, Good_j be the events that bad is never set in games G_i, G_j, respectively. Then,

$$\Pr\left[G_i^A \Rightarrow 1 \wedge \mathrm{Good}_i\right] = \Pr\left[G_j^A \Rightarrow 1 \wedge \mathrm{Good}_j\right].$$

In a multi-group signature scheme which prover has multiple secret keys, we consider two models of attackers. When multiple secret keys are required for generating of the signature, one attacker does not have any secret keys, called model 1, and another has a subset of the secret keys, called model 2.

Proof in the model 1

We first transform a forger F into an adversary A with the following properties. A has time-complexity $t(\cdot) + O(q_s)$, makes at most $1 + q_h(\cdot)$ hash queries, makes at most $q_s(\cdot)$ sign queries, has advantage no less than that of F, and additionally has the following properties:

(1) All of its hash queries are of the form $R \,\|\, Cmt \,\|\, m$ for some $R \in \{0,1\}^{c(k)}$ and Cmt, $m \in \{0,1\}^*$.
(2) Before outputting forgery $(m, R \,\|\, Cmt \,\|\, Rsp)$, the adversary A has made a hash query $R \,\|\, Cmt \,\|\, Rsp$.
(3) If A outputs $(m, R \,\|\, Cmt \,\|\, Rsp)$, then m was never a sign query.

We define an impersonator I against an ID. It has input pk

```
Game G_0
Initialize
000 (pk, sk) <-$ mgK(k); hc <- 0; sc <- 0
001 fp <-$ {1, ..., 1 + q_h(k)}
002 Ch* <-$ {0,1}^{c(k)}
003 For i = 1, ..., q_s(k) do
004     R_P^i <-$ Coins_P(k)
005     TCmt_i <- mgCo({sk}^N; R_P^i)
006     TCh_i <-$ {0,1}^{c(k)}
007     TRsp_i <- mgR({sk}^N, TCmt_i || TCh_i; R_P^i)
008 return pk

On H-query x
010 If HT[x] = ⊥ Then
011     hc <- hc + 1; QT[hc] <- x
012     If hc ≠ fp Then
013         y <-$ {0,1}^{c(k)}; HT[x] <- y
014     Else HT[x] <- Ch*
015 return HT[x]

On Sign-query M
020 sc <- sc + 1; R <-$ {0,1}^s
021 x <- R || TCmt_sc || m
022 HT[x] <- TCh_sc
023 return R || TCmt_sc || TRsp_sc

Finalize (M, σ)
030 Parse σ as R || Cmt || Rsp
031 Let i such that QT[i] = R || Cmt || m
032 If i ≠ fp Then bad <- true
033 return V({pk}^N, Cmt || Ch* || Rsp)
```

```
Game [G_1]/G_2
Initialize
100 (pk, sk) <-$ mgK(k); hc <- 0; sc <- 0
101 For i = 1, ..., q_s(k) do
102     R_P^i <-$ Coins_P(k)
103     TCmt_i <- mgCo({sk}^N; R_P^i)
104     TCh_i <-$ {0,1}^{c(k)}
105     TRsp_i <- mgR({sk}^N, TCmt_i || TCh_i; R_P^i)
106 return pk

On H-query x
110 If HT[x] = ⊥ Then
111     hc <- hc + 1; QT[hc] <- x
112     HT[x] <-$ {0,1}^{c(k)}
113 return HT[x]

On Sign-query M
120 sc <- sc + 1; R <-$ {0,1}^s
121 x <- R || TCmt_sc || m
122 HT[x] <- TCh_sc
123 return R || TCmt_sc || TRsp_sc

Finalize (M, σ)
130 Parse σ as R || Cmt || Rsp
131 Let i such that QT[i] = R || Cmt || m
132 Ch* <- HT[QT[i]]
133 fp <-$ {1, ..., 1 + q_h(k)}
134 If i ≠ fp Then
135     bad <- true [; Ch* <- HT[QT[fp]]]
136 return V({pk}^N, Cmt || Ch* || Rsp)
```

Fig. 1 Game G_0, G_1, and G_2

and access to a transcript oracle $\mathrm{Tr}^{ID}_{N,pk,sk,k}$. It begins with the initialization

$$hc \leftarrow 0;\ sc \leftarrow 0;\ fp \xleftarrow{\$} \{1,\ldots,1+q_s(k)\}$$
$$\text{For } i = 1,\ldots,q_s(k)$$
$$\text{do } TCmt_i \parallel TCh_i \parallel TRsp_i \xleftarrow{\$} \mathrm{Tr}^{ID}_{N,pk,sk,k}$$

Then, it runs A on input pk. We assume that A makes $q_h(k)$ hash-oracle queries and I will embed a challenge value in any return values of hash queries. When A makes a hash query x, the impersonator I returns $\mathrm{HT}[x]$ if this value is defined. Otherwise, it increments hc by one. If $hc \neq fp$, it simply picks $\mathrm{HT}[x]$ at random from $\{0,1\}^{c(k)}$ and returns it to A. Otherwise, it parses x as $R \parallel Cmt^* \parallel m$ and sends Cmt^* to the verifier as the output of $mgCo$. After it receives back a challenge Ch^*, it stores as $\mathrm{HT}[fp]$ and also returns to A as the response to hash query x. A cannot distinguish between $hc = fp$ or not. When A makes a sign query m, the impersonator I increments sc, picks R at random from $\{0,1\}^{s(k)}$, sets $\mathrm{HT}[R \parallel TCmt_{sc} \parallel m]$ to TCh_{sc} and returns $R \parallel TCmt_{sc} \parallel TRsp_{sc}$ to A as the signature. With the hash of $R \parallel TCmt_{sc} \parallel m$ defined as its TCh_{sc}, however, the signature is valid. Finally, A halts with output a forgery $(m, R\|Cmt\|Rsp)$. The impersonator I now send Rsp to the verifier as the output of mgR and halts. We claim that

$$\mathbf{Adv}^{\text{ima-pa}}_{mg\text{-}ID,I,N}(k) \geq \frac{1}{1+q_h(k)} \cdot \left(\mathbf{Adv}^{\text{euf-cma}}_{mg\text{-}DS,F,N} - \frac{[1+q_h(k)+q_s(k)]\cdot q_s(k)}{2^{s(k)+\beta(k)}}\right). \tag{2}$$

If Eq. (2) is true, then Eq. (1) is true.

We will use games G_0 to G_5 of Figs. 1 and 2 to derive Eq. (2) by rewriting the games. For $0 \leq i \leq 5$, let Good_i denote the event that game G_i never sets bad. We state a chain of inequalities which we will justify below:

$$\mathbf{Adv}^{\text{ima-pa}}_{mg\text{-}ID,I,N}(k) \geq \Pr\left[G_0^A \Rightarrow 1 \wedge \mathrm{Good}_0\right] \tag{3}$$
$$= \Pr\left[G_1^A \Rightarrow 1 \wedge \mathrm{Good}_1\right] \tag{4}$$
$$= \Pr\left[G_2^A \Rightarrow 1 \wedge \mathrm{Good}_2\right] \tag{5}$$
$$= \Pr\left[G_2^A \Rightarrow 1\right] \cdot \Pr[\mathrm{Good}_2] \tag{6}$$

Game G_0 simulates the execution environment of I. The interaction with the verifier is not explicit. Instead, the verifier's challenge Ch^* is chosen in line 002 of **Initialize**. I can obtain the transcript from its oracle, so line 004–007 generate their value. However, G_0 generates them explicitly by using the secret keys chosen at line 000. Parsing $\mathrm{QT}[fp]$ as $R \parallel Cmt^* \parallel m$, the value Cmt^* plays the role of the commitment sent by I to the verifier. If i, which generated at line 031, equals fp, then $I's$ conversation with the verifier is $\mathrm{Cmt} \parallel \mathrm{Ch}^* \parallel Rsp$, where $Cmt = Cmt^*$. Therefore, I succeeds when $mgV_A(pk, Cmt \parallel Ch^* \parallel Rsp) = 1$. We have justified Eq. (3).

In game G_0, Ch^* is picked up at random in **Initialize**.

Game $\boxed{G_3}/G_4$
Initialize
300 $(pk, sk) \xleftarrow{\$} mgK(k); hc \leftarrow 0; sc \leftarrow 0$
301 return pk

On H-query x
310 If $\mathrm{HT}[x] = \perp$ Then
311 $\quad hc \leftarrow hc + 1; \mathrm{QT}[hc] \leftarrow x$
312 $\quad \mathrm{HT}[x] \xleftarrow{\$} \{0,1\}^{c(k)}$
313 return $\mathrm{HT}[x]$

On Sign-query M
320 $sc \leftarrow sc + 1; R \xleftarrow{\$} \{0,1\}^{s}$
321 $R_P^i \xleftarrow{\$} \mathrm{Coins}_P(k)$
322 $TCmt_{sc} \leftarrow mgCo(\{sk\}^N; R_P^i); TCh_{sc} \xleftarrow{\$} \{0,1\}^{c(k)}$
323 $x \leftarrow R \parallel TCmt_{sc} \parallel m$
324 If $\mathrm{HT}[x] \neq \perp$ Then
325 $\quad$ **bad** $\leftarrow$ true $\boxed{; TCh_{sc} \leftarrow \mathrm{HT}[x]}$
326 $TRsp_{sc} \leftarrow mgR(\{sk\}^N, TCmt_{sc} \parallel TCh_{sc}; R_P^i)$
327 $\mathrm{HT}[x] \leftarrow TCh_{sc}$
328 return $R \parallel TCmt_{sc} \parallel TRsp_{sc}$

Finalize (M, σ)
330 Parse σ as $R \parallel Cmt \parallel Rsp$
331 Let i such that $\mathrm{QT}[i] = R \parallel Cmt \parallel m$
332 $Ch^* \xleftarrow{\$} \mathrm{HT}[\mathrm{QT}[i]]$
333 return $V(\{pk\}^N, Cmt \parallel Ch^* \parallel Rsp)$

Game G_5
Initialize
500 $(pk, sk) \xleftarrow{\$} mgK(k); hc \leftarrow 0; sc \leftarrow 0$
501 return pk

On H-query x
510 If $\mathrm{HT}[x] = \perp$ Then
511 $\quad hc \leftarrow hc + 1; \mathrm{QT}[hc] \leftarrow x$
512 $\quad \mathrm{HT}[x] \xleftarrow{\$} \{0,1\}^{c(k)}$
513 return $\mathrm{HT}[x]$

On Sign-query M
520 $sc \leftarrow sc + 1; R \xleftarrow{\$} \{0,1\}^{s}$
521 $R_P^i \xleftarrow{\$} \mathrm{Coins}_P(k)$
522 $TCmt_{sc} \leftarrow mgCo(\{sk\}^N; R_P^i)$
523 $x \leftarrow R \parallel TCmt_{sc} \parallel m$
524 If $\mathrm{HT}[x] \neq \perp$ Then $\mathrm{HT}[x] \xleftarrow{\$} \{0,1\}^{c(k)}$
525 $TCh_{sc} \xleftarrow{\$} \{0,1\}^{c(k)}$
526 $TRsp_{sc} \leftarrow mgR(\{sk\}^N, TCmt_{sc} \parallel TCh_{sc}; R_P^i)$
527 return $R \parallel TCmt_{sc} \parallel TRsp_{sc}$

Finalize (M, σ)
530 Parse σ as $R \parallel Cmt \parallel Rsp$
531 Let i such that $\mathrm{QT}[i] = R \parallel Cmt \parallel m$
532 $Ch^* \xleftarrow{\$} \mathrm{HT}[\mathrm{QT}[i]]$
533 return $V(\{pk\}^N, Cmt \parallel Ch^* \parallel Rsp)$

Fig. 2 Game G_3, G_4, and G_5

On the other hand, game G_1 does not choose it in **Initialize**, but instead assigns it the value HT[*fp*] in **Finalize**. Lines 132, 134, and 135 do this because the boxed code is included in G_1. Since *fp* is not used in returning as output of hash queries, G_1 delays its choice until line 133. Thus, this explains Eq. (4).

Games G_1, G_2 are *identical until* **bad** games, so Eq. (5) is implied from Eq. (4) and Lemma 2. The difference between games is whether the boxed statement at line 135 exists or not. Since *fp* is not used in determining the game output, the events Good_2 and $G_2^A \Rightarrow 1$ are independent, justifying Eq. (6).

From lines 133-135 of G_2, it is clear that

$$\Pr[\mathrm{Good}_2] = 1/\{1 + q_h(k)\}.$$

The **Finalize** procedure of G_3 has the same output as that of G_2. However, lines 133-135 are absent in G_3. The other change it makes is to delay the choices of lines 101-105 until they are needed in replaying sign queries. These replies are the same as in G_2. The setting of **bad** does not affect the game output, so we have

$$\Pr\left[G_2^A \Rightarrow 1\right] = \Pr\left[G_3^A \Rightarrow 1\right] \tag{7}$$
$$\geq \Pr\left[G_4^A \Rightarrow 1\right] - \Pr\left[G_4^A \text{ sets } \mathbf{bad}\right] \tag{8}$$

where Eq. (8) is obtained from Eq. (7) and Lemma 1 since games G_3, G_4 are *identical until* **bad** games. When the value x of line 323 had been provided, G_4 sets **bad**. Therefore, the probability that the i-th sign query sets **bad** in G_4 is at most

$$\{1 + q_h(k) + (i + 1)\} \Big/ \left\{2^{s(k)+\beta(k)}\right\}.$$

So,

$$\begin{aligned} &\Pr\left[G_4^A \text{ sets } \mathbf{bad}\right] \\ &\leq \sum_{i=1}^{q_s(k)} \frac{1 + q_h(k) + (i+1)}{2^{s(k)+\beta(k)}} \\ &= \frac{q_h(k)q_s(k) + \dfrac{q_s(k)(q_s(k)+1)}{2}}{2^{s(k)+\beta(k)}} \\ &\leq \frac{[1 + q_h(k) + q_s(k)]q_s(k)}{2^{s(k)+\beta(k)}}. \end{aligned} \tag{9}$$

Given that the boxed code of line 325 is not present in G_4, the code to reply to sign queries is equivalent to that in G_5 barring to longer setting **bad**. The latter does not affect the game output, so

$$\Pr\left[G_4^A \Rightarrow 1\right] = \Pr\left[G_5^A \Rightarrow 1\right].$$

But G_5 captures the experiment defining the advantage of A and so

$$\Pr\left[G_5^A \Rightarrow 1\right] = \mathbf{Adv}_{mg-DS,A,N}^{\mathrm{euf-cma}}(k) \tag{10}$$
$$\geq \mathbf{Adv}_{mg-DS,F,N}^{\mathrm{euf-cma}}(k) \tag{11}$$

the last by the properties of A stated above. Putting together Eqs. (3)–(11) yields Eq. (2). ■

Proof in the model 2

When N secret keys are required for generating of the signature, we assume that an adversary has $N - L$ secret keys. If the output of response from N secret keys and that from $N - L$ secret keys are independent, it is necessary for the adversary to obtain L secret keys which the adversary does not have. In other words,

$$\begin{aligned}&\mathbf{Adv}^{\mathrm{euf-cma}}_{mg-DS,F,N}(k)\ (\textit{in model } 1)\\ &\quad\le \mathbf{Adv}^{\mathrm{euf-cma}}_{mg-DS,F,N}(k)\ (\textit{in model } 2)\\ &\quad= \mathbf{Adv}^{\mathrm{euf-cma}}_{mg-DS,F,L}(k)\ (\textit{in model } 1)\end{aligned}$$

Thus, model 2 is can considered in the same as the model 1. ■

5. Concrete Construction

This section presents a concrete construction of multi-group signature scheme which is EUF-CMA secure based on a code-based multi-group authentication scheme [1] which follows definition in Sect. 2.4 and is IMP-PA secure.

5.1 Algorithms

The concrete scheme is composed of three algorithms, **KeyGen**, **Sign**, and **Verify**.

KeyGen$(1^n, M)$**:** It takes as input the security parameter n. It selects a random binary n-tuple $s_i \in \mathbb{F}_2^n$ with Hamming weight $\omega_i = \mathrm{wt}(s_i)$ as the secret key $k_s^{(i)}$ and a triple comprising a random binary parity-check matrix H, the syndrome $p_i = Hs_i^T$, and ω_i as the corresponding public key $k_p^{(i)}$ of the secret key $k_s^{(i)}$. It outputs the secret/public key pair $\left(k_s^{(i)}, k_p^{(i)}\right)$. A prover has M key pairs by generating key pairs M times.

Sign$\left(\left\{k_s^{(i)}\right\}_{i=1}^M, m\right)$**:** To sign a message $m \in \{0,1\}^*$, it runs the following steps:

1. It picks a random n-bits word $y_j \in_R \mathbb{F}_2^n$ together with a random permutation σ_j of the integers $\{1\cdots n\} \in_R S_n$.
2. By using y_j, σ_j, it computes $\mathrm{CMT}^j = \left(c_1^{(j)}, c_2^{(j)}, c_3^{(j)}\right)$ as follows:
$$\begin{cases} c_1^{(j)} = h(\sigma_j, Hy_j^T) \\ c_2^{(j)} = h(\sigma_j(y_j)) \\ c_3^{(j)} = h\left(\sigma_j\left(y_j + \sum_{i=1}^M s_i\right)\right)\end{cases}$$
3. It repeats Step 1 and Step 2 r times and obtains $\mathrm{CMT} = \{\mathrm{CMT}^j\}^r = (\mathrm{CMT}^1, \ldots, \mathrm{CMT}^r)$.
4. It hashes m and CMT as follows:
$$\mathrm{Ch}^j = h'(m, \mathrm{CMT}^j) \in_R \{0,1,2\}$$
It obtains $\mathrm{Ch} = \{\mathrm{Ch}^j\}^r = (\mathrm{Ch}^1, \ldots, \mathrm{Ch}^r)$.
5. It selects RSP^j corresponding to Ch^j as follows:
If $\mathrm{Ch}^j = 0$: $\mathrm{RSP}^j := (y_j, \sigma_j)$.
If $\mathrm{Ch}^j = 1$: $\mathrm{RSP}^j := (y_j + \sum_{i=1}^M s_i, \sigma_j)$.
If $\mathrm{Ch}^j = 2$: $\mathrm{RSP}^j := \left(\sigma_j(y_j), \{\sigma_j(s_i)\}_{i=1}^M\right)$.
6. Then, it outputs a signature $\Sigma = (\mathrm{CMT}^1, \ldots, \mathrm{CMT}^r; (\mathrm{Ch}^1, \ldots, \mathrm{Ch}^r); \mathrm{RSP}^1, \ldots, \mathrm{RSP}^r)$.

Verify$\left(\left\{k_p^{(i)}\right\}_{i=1}^M, m, \Sigma\right)$: To verify that the signature Σ is collect, it run the following steps:

1. If $(\mathrm{Ch}^1, \ldots, \mathrm{Ch}^r) \neq h'(m; \mathrm{CMT}^1, \ldots, \mathrm{CMT}^r)$ then it regards that Σ is not collect and returns ***reject***.
Otherwise, it proceeds Step 2.
2. For $j = 1, \ldots, r$, it verifies RSP^j corresponding to Ch^j and CMT^j as follows:
If $\mathrm{Ch}^j = 0$: The verifier checks that $c_1^{(j)}$, $c_2^{(j)}$, which were made in step 2, have been computed honestly. The equations to check are as follows:
$$\begin{cases} c_1^{(j)} = h(\sigma_j, Hy_j^T) \\ c_2^{(j)} = h(\sigma_j(y_j))\end{cases}$$
If $\mathrm{Ch}^j = 1$: The verifier checks that $c_1^{(j)}$, $c_3^{(j)}$ were correct. The equations to check are as follows:
$$\begin{cases} c_1^{(j)} = h\left(\sigma_j, H\left(y_j + \sum_{i=1}^M s_i\right)^T + \sum_{i=1}^M p_i\right) \\ c_3^{(j)} = h\left(\sigma_j\left(y_j + \sum_{i=1}^M s_i\right)\right)\end{cases}$$
If $\mathrm{Ch}^j = 2$: The verifier checks the weight property and $c_2^{(j)}$, $c_3^{(j)}$. The equations to check are as follows:
$$\begin{cases} c_2^{(j)} = h(\sigma_j(y_j)) \\ c_3^{(j)} = h\left(\sigma_j(y_j) + \sum_{i=1}^M \sigma_j(s_i)\right) \\ \{\mathrm{wt}(\sigma_j(s_i)) = \omega_i\}_{i=1}^M\end{cases}$$
3. When it doesn't return ***reject*** in Step 1 or 2, it regards Σ as the valid signature of m and returns ***accept***.

5.2 Correctness of Algorithms

A signature made by **Sign** must be accepted by **Verify**. For $j = 1, \ldots, r$, it can be shown as follows:
If $\mathrm{Ch}^j = 0$: **Verify** can compute the values of $h(\sigma_j, Hy_j^T)$ and $h(\sigma_j(y_j))$ since RSP^j contains (y_j, σ_j). The signature is accepted when the values of $h(\sigma_j, Hy_j^T)$ and $h(\sigma_j(y_j))$ are equal to $c_1^{(j)}$ and $c_2^{(j)}$, respectively.
If $\mathrm{Ch}^j = 1$: **Verify** can compute the values of $h\left(\sigma_j, H\left(y_j + \sum_{i=1}^M s_i\right)^T + \sum_{i=1}^M p_i\right)$ and $h\left(\sigma_j\left(y_j + \sum_{i=1}^M s_i\right)\right)$ since RSP^j contains $\left(y_j + \sum_{i=1}^M s_i, \sigma_j\right)$. From the syndrome p_i is Hs_i^T, then we have

$$\begin{aligned}&h\left(\sigma_j, H\left(y_j + \sum_{i=1}^M s_i\right)^T + \sum_{i=1}^M p_i\right)\\ &= h\left(\sigma_j, Hy_i^T + H\left(\sum_{i=1}^M s_i\right)^T + \sum_{i=1}^M p_i\right)\\ &= h\left(\sigma_j, Hy_i^T + \sum_{i=1}^M Hs_i^T + \sum_{i=1}^M p_i\right)\\ &= h(\sigma_j, Hy_i^T).\end{aligned}$$

The signature is accepted when the values of $h\Big(\sigma_j, H\Big(y_j +$

$\sum_{i=1}^{M} s_i\big)^T + \sum_{i=1}^{M} p_i\big)$ and $h\big(\sigma_j\big(y_j + \sum_{i=1}^{M} s_i\big)\big)$ are equal to $c_1^{(j)}$ and $c_3^{(j)}$, respectively.

If $\mathrm{Ch}^j = 2$: **Verify** can compute the values of $h(\sigma_j(y_j))$, $h\big(\sigma_j(y_j) + \sum_{i=1}^{M} \sigma_j(s_i)\big)$ and $\big\{\mathrm{wt}(\sigma_j(s_i))\big\}_{i=1}^{M}$ since RSP^j contains $\big(\sigma_j(y_j), \{\sigma_j(s_i)\}_{i=1}^{M}\big)$. From σ_j is a random permutation, Hamming weights of $\sigma_j(s_i)$ and s_i are the same value. Then we have

$$h\Big(\sigma_j(y_j) + \sum\nolimits_{i=1}^{M} \sigma_j(s_i)\Big) = h\Big(\sigma_j(y_i) + \sigma_j\Big(\sum\nolimits_{i=1}^{M} s_i\Big)\Big)$$
$$= \Big(\sigma_j\Big(y_i + \sum\nolimits_{i=1}^{M} s_i\Big)\Big).$$

The signature is accepted when the values of $h(\sigma_j(y_j))$, $h\big(\sigma_j(y_j) + \sum_{i=1}^{M} \sigma_j(s_i)\big)$ and $\big\{\mathrm{wt}(\sigma_j(s_i))\big\}_{i=1}^{M}$ are equal to $c_2^{(j)}$, $c_3^{(j)}$ and $\{\omega_i\}_{i=1}^{M}$.

6. Conclusion

This paper presented how to construct multi-group signature schemes from multi-group authentication schemes and gave its security proof. A concrete multi-group signature scheme which is EUF-CMA secure from a code-based multi-group authentication scheme which is IMP-PA secure by using our transform was shown. We will select the appropriate parameters and consider how to construct multi-group signature schemes from a signature scheme which uses a single key in future work.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 16K00184 and 22700067.

References

[1] T.R. Halford, "How to Prove Yourself to Multiple Parties: Energy-Efficient Multi-group Authentication," Proc. IEEE MILCOM 2013, pp.237–242, Nov. 2013.
[2] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," Advances in Cryptology-CRYPTO'86, pp.186–194, Dec. 1986.
[3] K.H.M. Wong, Y. Zheng, J. Cao, and S. Wang, "A dynamic user authentication scheme for wireless sensor networks," Proc. IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06), vol.1, pp.244–251, June 2006.
[4] M.L. Das, "Two-factor user authentication in wireless sensor networks," IEEE Trans. Wireless Commun., vol.8, no.3, pp.1086–1090, May 2009.
[5] M.K. Khan and K. Alghathbar, "Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks'," Sensors, vol.10, no.3, pp.2450–2459, March 2010.
[6] B. Vaidya, D. Makrakis, and H.T. Mouftah, "Improved two-factor user authentication in wireless sensor networks," Proc. IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp.600–606, Oct. 2010.
[7] K. Ren, S. Yu, W. Lou, and Y. Zhang, "Multi-user broadcast authentication in wireless sensor networks," IEEE Trans. Veh. Technol., vol.58 no.8, pp.4554–4564, Oct. 2009.
[8] N. Bruce and H.J. Lee, "Cryptographic computation of private shared key based mutual authentication protocol: Simulation and modeling over wireless networks," Proc. The International Conference on Information Networking 2014 (ICOIN2014), pp.578–582, Feb. 2014.
[9] M.K. Sharma, R.S. Bali, and A. Kaur, "Dyanimc key based authentication scheme for Vehicular Cloud Computing," Proc. 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), pp.1059–1064, Oct. 2015.
[10] S.K. Udgata, A. Mubeen, and S.L. Sabat, "Wireless sensor network security model using zero knowledge protocol," Proc. International Conference on Communications (ICC), pp.1–5, June 2011.
[11] J. Stern, "A new paradigm for public key identification," IEEE Trans. Inf. Theory, vol.42, no.6, pp.1757–1768, Nov. 1996.
[12] P. Véron, "Improved identification schemes based on error-correcting codes," Applicable Algebra in Engineering, Communication and Computing, vol.8, no.1, pp.57–69, Jan. 1997.
[13] P. Gaborit and M. Girault, "Lightweight code-based identification and signature," Proc. IEEE International Symposium on Information Theory, pp.191–195, June 2007.
[14] C. Aguilar, P. Gaborit, and J. Schrek, "A new zero-knowledge code based identification scheme with reduced communication," CoRR, abs/1111.1644, 2011.
[15] A. Dambra, P. Gaborit, M. Roussellet, J. Schrek, and N. Tafforeau, "Improved Secure Implementation of Code-Based Signature Schemes on Embedded Devices," IACR Cryptology ePrint Archive: Report 2014/163, March 2014.
[16] R. Ghanbarimaman and A.N. Pour, "A new definition of group authentication increasing performance of server calculation," Proc. 2012 International Conference on Information Science and Applications (ICISA), pp.1–6, May 2012.
[17] L. Harn, "Group authentication," IEEE Trans. Comput., vol.62, no.9, pp.1893–1898, Sept. 2013.
[18] C. Guo, R. Zhuang, L. Yuan, and B. Feng, "A Group Authentication Scheme Supporting Cheating Detection and Identification," Proc. 2015 Ninth International Conference on Frontier of Computer Science and Technology (FCST), pp.110–114, Aug. 2015.
[19] U. Feige, A. Fiat, and A. Shamir, "Zero-knowledge proofs of identity," Journal of cryptology, vol.1, no.2 pp.77–94, June 1988.
[20] S. Goldwasser, S. Micali, and R.L. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," SIAM Journal on Computing, vol.17, no.2, pp.281–308, 1988.
[21] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," Proc. 1st ACM conference on Computer and communications security (CCS'06), pp.62–73, Dec. 1993.
[22] D. Pointcheval and J. Stern, "Security Arguments for Digital Signatures and Blind Signatures," Journal of Cryptology, vol.13, no.3, pp.361–396, 2000.
[23] M. Abdalla, J.H. An, M. Bellare, and C. Namprempre, "From Identification to Signatures Via the Fiat-Shamir Transform: Necessary and Sufficient Conditions for Security and Forward-Security," IEEE Trans. Inf. Theory, vol.54, no.8, pp.3631–3646, 2008. (Conference Ver.: Proc. EUROCRYPT 2002, LNCS, vol.2332, pp.418–433, 2002).
[24] B. Chor and O. Goldreich, "Unbiased bits from sources of weak randomness and probabilistic communication complexity," SIAM Journal on Computing, vol.17, no.2, pp.230–261, 1988.
[25] M. Bellare and P. Rogaway, "The security of triple encryption and a framework for code-based game-playing proofs," Advances in Cryptology-EUROCRYPT 2006, pp.409–426, May 2006.

Kenta Nomura received B.E. degree from Kobe University, Japan, 2015. He is currently master course student at Kobe University. His current research interests include information security and cryptography.

Masami Mohri received B.E. and M.E. degrees from Ehime University, Japan, in 1993 and 1995 respectively. She received Ph.D degree in Engineering from the University of Tokushima, Japan in 2002. From 1995 to 1998 she was an assistant professor at the Department of Management and Information Science, Kagawa junior college, Japan. From 1998 to 2002 she was a research associate of the Department of Information Science and Intelligent Systems, the University of Tokushima, Japan. From 2003 to 2008 she was a lecturer of the same department. Since 2008, she has been an associate professor at the Information and Multimedia Center, Gifu University, Japan. Her research interests are in coding theory, information security and cryptography. She is a member of IEEE.

Yoshiaki Shiraishi received B.E. and M.E. degrees from Ehime University, Japan, and Ph.D degree from the University of Tokushima, Japan, in 1995, 1997, and 2000, respectively. From 2002 to 2006 he was a lecturer at the Department of Informatics, Kinki University, Japan. From 2006 to 2013 he was an associate professor at the Department of Computer Science and Engineering, Nagoya Institute of Technology, Japan. Since 2013, he has been an associate professor at the Department of Electrical and Electronic Engineering, Kobe University, Japan. His current research interests include information security, cryptography, computer network, and knowledge sharing and creation support. He received the SCIS 20th Anniversary Award and the SCIS Paper Award from ISEC group of IEICE in 2003 and 2006, respectively. He received the SIG-ITS Excellent Paper Award from SIG-ITS of IPSJ in 2015. He is a member of IEEE, ACM, and a senior member of IPSJ.

Masakatu Morii received the B.E. degree in electrical engineering and the M.E. degree in electronics engineering from Saga University, Saga, Japan, and the D.E. degree in communication engineering from Osaka University, Osaka, Japan, in 1983, 1985, and 1989, respectively. From 1989 to 1990 he was an Instructor in the Department of Electronics and Information Science, Kyoto Institute of Technology, Japan. From 1990 to 1995 he was an Associate Professor at the Department of Computer Science, Faculty of Engineering, Ehime University, Japan. From 1995 to 2005 he was a Professor at the Department of Intelligent Systems and Information Science, Faculty of Engineering, the University of Tokushima, Japan. Since 2005, he has been a Professor at the Department of Electrical and Electronic Engineering, Faculty of Engineering, Kobe University, Japan. His research interests are in error correcting codes, cryptography, discrete mathematics and computer networks and information security. He is a member of the IEEE.