Efficient and Unconditionally Secure Digital Signatures and a Security Analysis of a Multireceiver Authentication Code

Goichiro Hanaoka[1], Junji Shikata[1], Yuliang Zheng[2], and Hideki Imai[1]

[1] Information & Systems, Institute of Industrial Science, University of Tokyo
4-6-1 Komaba, Meguro-ku, Tokyo 153-8508, Japan.
{hanaoka,shikata}@imailab.iis.u-tokyo.ac.jp, imai@iis.u-tokyo.ac.jp
[2] Department of Software and Information Systems, UNC Charlotte
9201, University City Blvd, Charlotte, NC 28223, USA
yzheng@uncc.edu

Abstract. Digital signatures whose security does not rely on any unproven computational assumption have recently received considerable attention. While these unconditionally secure digital signatures provide a foundation for long term integrity and non-repudiation of data, currently known schemes generally require a far greater amount of memory space for the storage of users' secret information than a traditional digital signature. The focus of this paper is on methods for reducing memory requirements of unconditionally secure digital signatures. A major contribution of this paper is to propose two novel unconditionally secure digital signature schemes that have significantly shortened secret information for users. As a specific example, with a typical parameter setting the required memory size for a user is reduced to approximately $\frac{1}{10}$ of that in previously known schemes. Another contribution of the paper is to demonstrate an attack on a multireceiver authentication code proposed by Safavi-Naini and Wang, and present a method to fix the problem of the code.

1 Introduction

Digital signatures represent one of the most widely used security technologies for ensuring unforgeability and non-repudiation of digital data. While some data only require the assurance of integrity for a relatively short period of time, say up to two years, there are many cases where it is necessary for signed documents to be regarded as legally valid for a much longer period of time. Some of the examples of data that require long-term integrity include court records, long-term leases and contracts.

In August 1999, a team of cryptography researchers from around the world completed the factorization of an 512-bit RSA composite with the use of the Number Field Sieve method [3]. With the rapid advancement in the speed of computers, one can safely predict that factoring even larger composites may become feasible at some point of time in future. We also note that innovative

D. Naccache and P. Paillier (Eds.): PKC 2002, LNCS 2274, pp. 64–79, 2002.

factoring algorithms may emerge, dramatically changing the landscape of public key cryptosystems whose security hinges on the presumed hardness of certain number theoretic problems. In yet another significant development, the past few years have witnessed significant progress in quantum computers. These computers, if built, will have the capacity to improve profoundly known algorithms for factoring and solving discrete logarithms [16,1], whereby challenging the long term security of all digital signature schemes based on number-theoretic problems.

The above discussions show clearly that there is a need to devise digital signature schemes that provide assurance of long term integrity. A possible solution to this problem is digital signature schemes whose security does not rely on any unproven assumption. The present authors have recently proposed the first unconditionally secure digital signature schemes (with transferability) [9]. An interesting and very useful property of these signature schemes is that they admit transferability, allowing the recipient of a signature to transfer it to another recipient without fearing that the security of the signature might be compromised. However, these signature schemes do have a disadvantage, namely the size of a user's secret information is very large. This disadvantage may pose a serious problem in practice, especially when a user's secret information need to be stored in such devices as smart cards.

A major contribution of this work is to propose two novel unconditionally secure digital signature schemes that require significantly less amount of memory for each user's secret information. As an example, consider an organization that has 100,000 users. With the new signature schemes, the required memory size for each user is reduced to approximately $\frac{1}{10}$ of that required by previously known schemes. Another contribution of this paper is to present an attack on a multireceiver authentication code proposed by Safavi-Naini and Wang, which is followed by a method to fix that problem. Safavi-Naini and Wang's multireceiver authentication code is related to one of our new unconditionally secure digital signature schemes. More specifically, one of our approaches succeeds in reducing the required memory size for a user's secret information by unifying secret data for both signing and verification.

1.1 Related Work

Unconditionally secure authentication codes. There have been attempts to modify unconditionally secure authentication codes [7,17] with the aim of enhancing the codes with added security properties. An obvious approach is to transform an unconditionally secure authentication code into an unconditionally secure digital signature. To achieve this, however, one faces two insurmountable technical hurdles. The first hurdle lies in authentication codes, especially the conventional Cartesian ones, which do not provide the function of non-repudiation, simply because a receiver can easily forge a sender's message and vice versa. The second hurdle is that the receiver is always designated, which means that a signature cannot be verified by another party without having the shared key.

An extension to authentication codes is called, *authentication codes with arbitration* or A^2-codes [18,19,10,8]. These codes involve a trusted third party called an arbiter. The arbiter help resolve disputes at times when a receiver forges a sender's message or the sender claims that the message has been forged by the receiver. A^2-codes have been further improved to have a less trustworthy arbiter as one of the requirements. These improved codes are called, A^3-codes [2,5,8]. A property common to both codes is that the receiver of an authenticated message has to be designated. Therefore, in a signature system where the receiver is not designated, both A^2-codes and A^3-codes cannot be used as digital signatures.

Another extension made to authentication codes, *multireceiver authentication codes* (MRA) [6,13,8], have been extensively studied in the literature. In a MRA scheme, a broadcast message can be verified by any one of the receivers. Earlier MRA schemes required the sender himself to be designated. In order to ease the requirement of the designated sender, several variations of *MRA with dynamic sender* or DMRA have been proposed [13,14,15]. Among these schemes, we especially looked into Safavi-Naini and Wang's DMRA [13,15] which we thought has an interesting construction. In their scheme, a user's secret information for generating authenticated messages and that for verifying them is the same. Which means that, their scheme requires significantly less amount of memory size compared to other DMRAs. Further, in one of our new schemes, with this application, the required memory size for a user's secret information of our schemes can be reduced as well.

It is important to note that these schemes make sense only in the case of broadcasting. If MRA or DMRA is used for point-to-point authentication, then the sender can easily generate a fraudulent message, which is accepted by the receiver and not by other participants. The situation is made complex due to a reason that the same fraudulent message may had been generated by the receiver himself. A further problem associated to this situation is that, MRA nor DMRA provide transferability. In particular, if an authenticated message is transferred from one verifier to another, the second verifier can forge a message that appears to be perfectly valid and may naturally transfer it to the next verifier. For these reasons, neither MRA nor DMRA satisfies the non-repudiation requirement of digital signature.

Unconditionally secure digital signatures. Chaum and Roijakkers [4] originally made the attempt to construct an unconditionally secure signature scheme using cryptographic protocols. However, their basic scheme was impractical, as it only signed a single bit message. Furthermore, their level of security of a signature decreased as the signature moved from one verifier to another. In practice, it is important for a signature scheme to have *transferability*, i.e., its security is not compromised when a signature is transferred among users. By applying A^3-codes, Johansson [8] proposed an improved version of Chaum-Roijakkers scheme, but Johansson did not address transferability of signature scheme.

Pfitzmann and Waidner proposed another version of unconditionally secure signature schemes [11,12]. However, their unconditional security was limited for signers. Recently, the present authors proposed an unconditionally secure digital

signature which addresses all known required properties including transferability [9]. However, that signature scheme (the HSZI-AC00 scheme, for short) requires a large amount of memory, which could be a problem in certain applications, e.g. smart card based systems.

1.2 Main Results

In this paper, we first present an attack on Safavi-Naini and Wang's DMRA [15]. More specifically, in their scheme, by observing a valid signature of an honest signer, a coalition of adversaries can make an impersonation attack with non-negligible probability. We also show a simple method to fix that problem.

Next, we show two novel unconditionally secure digital signature schemes that admit transferability. Both these schemes significantly reduce the required memory size for a user's secret information. In the first one, *symmetric construction*, the required memory size for a user's secret information is significantly reduced by unifying secret information for signing and that for verification. However, the required memory size for a signature is slightly increased compared to the HSZI-AC00 scheme. The basic idea behind unifying secret information for signing and verification in the symmetric construction is partially based on the idea from the fixed version of Safavi-Naini and Wang's DMRA. In the second construction, *asymmetric construction*, the required memory size is reduced without increasing the required memory size for a signature. More precisely, this scheme is optimal in terms of the required memory size for a signature as well as in the HSZI-AC00 scheme. As an example for 100,000 users with appropriate security parameter settings, the required memory size for a user is reduced to $\frac{1}{10}$ of that required in the previous method.

The organization of the remaining part of this paper is as follows: In Section 2, we give a brief review of Safavi-Naini and Wang's multireceiver authentication code, and demonstrate an attack on it. We also show a method to fix the problem. In Section 3, new unconditionally secure digital signature schemes are presented. Lastly, Section 4 presents a comparison between the proposed schemes with the previous method.

2 An Analysis of Safavi-Naini and Wang's DMRA

In general, DMRA is an authentication code where any entity in a system can generate and verify an authenticated message. In this section, we give a brief review of Safavi-Naini and Wang's multireceiver authentication codes with dynamic senders (the SW-DMRA, for short) [13,15]. As already mentioned, in this scheme, secret information for generating an authenticated message and that for verifying is the same. Primarily due to this property, the required memory size for a user's secret information in the SW-DMRA could be decreased to be significantly smaller to that of other DMRAs. However, the SW-DMRA is insecure when used as in [15]. In this section, we also demonstrate an attack on the SW-DMRA, and present a method to fix that problem. This attack is easy to

perform and indeed, very effective. In this attack, by observing a valid authenticated message, colluders can forge any user's valid authenticated message with probability 1.

U_j accepts the broadcasted message if $f_i(U_j) = f_j(U_i)$.

2.1 Implementation of Safavi-Naini and Wang's DMRA

In this subsection, the construction of the SW-DMRA is shown in more detail. This scheme was originally presented in [13] and was then improved and simplified in [15]. Here, we show the improved version. The model of DMRA follows [15].

Let F_q be the finite field with q elements and $\mathcal{S}$ the set of source states. We assume $\mathcal{S} = F_q$ and that each user's identity U_i is represented as distinct number on F_q, and ω is the maximum number of colluders in the system. The construction of the SW-DMRA is as follows.

Safavi-Naini and Wang's DMRA [15]

1. **Key distribution:** The TA chooses uniformly at random two symmetric polynomials $F_0(x, y)$ and $F_1(x, y)$ over F_q with two variables x and y of degree less than $\omega + 1$.[1] For each U_i $(i = 1, \cdots, n)$, the TA privately sends a pair of polynomial $\{F_0(x, U_i), F_1(x, U_i)\}$ to U_i. This constitutes the secret information of U_i.
2. **Broadcast:** If U_i wants to authenticate a source state $s \in F_q$, U_i calculates the polynomial $a_i(x) := F_0(x, U_i) + sF_1(x, U_i)$ and broadcasts $(s, a_i(x))$ with his identity to other users.
3. **Verification:** U_j can verify the authenticity of $(s, a_i(x))$ by first calculating the polynomial $b_j(x) := F_0(x, U_j) + sF_1(x, U_j)$ and then accepting $(s, a_i(x))$ as authentic and being sent from U_i if $b_j(U_i) = a_i(U_j)$.

2.2 Performance

As shown in above, in this scheme, U_i's secret information $\{F_0(x, U_i), F_1(x, U_i)\}$ is utilized for both generating and verifying authenticated message. Namely, for each user, the whole distributed secret information is used whether he is a sender or a recipient. Hence, the required memory size for a user's secret information can be reduced to significantly small value. More precisely, this scheme is optimal in terms of the required memory size for a user's secret information due to lower

[1] It is important to note that the meaning of the parameter ω in this paper is different from that of w used in [15]. The authors of [15] describe "no $w-1$ subset of users can perform impersonation and/or substitution attack on any other pair of users" ([15], page 161, Def. 5.1) and "Then TA randomly chooses two symmetric polynomials of degree less than w with coefficients in $GF(q)$" ([15], page 163). Thus, we can see that ω in this paper is equivalent to $w - 1$ in [15]. We also note that our definition of ω is in line with relevant papers by other researchers, including [6,8].

bound on it [15]. In addition, this scheme is also optimal in terms of the required memory size for an authenticated message [15]. For the details, see Theorem 5.2 in [15].

Although the authors of [15] claimed that the probability of succeeding for a collusion of up to ω users in performing all known attacks is at most $\frac{1}{q}$, however, the above scheme is insecure. The details regarding the security of this scheme is shown in [15]. In the next section, we demonstrate an attack on the above DMRA.

Here, we further point out the transferability of DMRAs. Generally in DMRAs as already mentioned, messages are transmitted over a broadcast channel, and in this particular situation, transferability is not required. However, for a digital signature (for point-to-point communication), transferability is a property that cannot be neglected. That is, a signature system must allow users to pass signatures among users without compromising the integrity of them. Generally speaking, DMRAs (and MRAs) do not fulfill this requirement. As an example to this, we show the vulnerability of the above DMRA where it allows users to pass authenticated messages among users without a broadcast channel.

Suppose that, U_{i_0} generates $(s, a_{i_0}(x))$ and sends it to U_{i_1}. Then, an adversary can modify the authenticated message as $(s, a'_{i_0}(x))$, such that $a'_{i_0}(U_{i_1}) = a_{i_0}(U_{i_1})$ and $a'_{i_0}(U_{i_2}) \neq a_{i_0}(U_{i_2})$ for a certain user U_{i_2}. On receiving $(s, a'_{i_0}(x))$, U_{i_1} accepts it as valid since $a'_{i_0}(U_{i_1}) = b_{i_1}(U_{i_0})$. However, when U_{i_1} further transfers $(s, a'_{i_0}(x))$ to U_{i_2}, U_{i_2} does not accept it since $a'_{i_0}(U_{i_2}) \neq b_{i_2}(U_{i_0})$, and U_{i_1} will be suspected to have forged it. We call this type of attack *transfer with a trap* following to [9]. For this reason, DMRA (and MRA) cannot be used as a digital signature.

In the remaining part of this section, we show an attack on the SW-DMRA, and also present a method to fix that problem. This attack is easy to perform and indeed, very effective. In this attack, by observing a valid authenticated message, ω colluders can forge any user's valid authenticated message with probability 1.

2.3 Attack on Safavi-Naini and Wang's DMRA

Let $\mathcal{W} = \{U_1, \cdots, U_\omega\}$ be the set of the colluders. These colluders can forge any user's authenticated message as described. When $U_0 (\notin \mathcal{W})$ transmits a valid authenticated message $(s, a_0(x))$, the colluders interrupt it and use it for forgery of another user's authenticated message. On observing $(s, a_0(x))$, the colluders generate authenticated messages $(s, a_1(x)), (s, a_2(x)), \cdots, (s, a_\omega(x))$. Letting

$$F_l(x, y) := (1, x, x^2, \cdots, x^\omega) A_l \begin{pmatrix} 1 \\ y \\ y^2 \\ \vdots \\ y^\omega \end{pmatrix}, \quad l = 0, 1,$$

where A_l $(l = 0, 1)$ are $(\omega+1)\times(\omega+1)$ symmetric matrices over F_q, the colluders now have a matrix D, where

$$D := (A_0 + sA_1) \begin{pmatrix} 1 & 1 & \cdots & 1 \\ U_0 & U_1 & \cdots & U_\omega \\ {U_0}^2 & {U_1}^2 & \cdots & {U_\omega}^2 \\ \vdots & \vdots & \cdots & \vdots \\ {U_0}^\omega & {U_1}^\omega & \cdots & {U_\omega}^\omega \end{pmatrix}.$$

Then, by using D, $A_0 + sA_1$ can be easily obtained as follows:

$$A_0 + sA_1 = D \begin{pmatrix} 1 & 1 & \cdots & 1 \\ U_0 & U_1 & \cdots & U_\omega \\ {U_0}^2 & {U_1}^2 & \cdots & {U_\omega}^2 \\ \vdots & \vdots & \cdots & \vdots \\ {U_0}^\omega & {U_1}^\omega & \cdots & {U_\omega}^\omega \end{pmatrix}^{-1}.$$

If the colluders $\mathcal{W}$ want to forge an authenticated message of a user U_j, where $U_j \notin \mathcal{W} \cup \{U_0\}$, $\mathcal{W}$ calculate

$$a'_j(x) = (1, U_j, U_j^2, \cdots, U_j^\omega)(A_0 + sA_1),$$

and broadcast $(s, a'_j(x))$ as an authenticated message of U_j for the source state s. Since $(s, a'_j(x))$ is exactly equal to U_j's valid authentication message for source state s, the colluders succeed in impersonation (or entity substitution) for U_j (with probability 1).

2.4 Method to Fix the Problem

An essential problem in the SW-DMRA is that $A_0 + sA_1$ can be calculated by using both ω colluders' secret information and an authenticated message generated by an honest user. In order to avoid calculating $A_0 + sA_1$, the rank of $A_0 + sA_1$ must be larger than ω. This implies that the degree of x and y in $F_0(x, y)$ and $F_1(x, y)$ must be at least $\omega + 1$. Letting the degree of x and y in $F_0(x, y)$ and $F_1(x, y)$ be at least $\omega + 1$, the colluders cannot succeed in the above attack with non-negligible probability. (See also the footnote that appeared earlier in this paper regarding the small but subtle difference between the definition of ω in this paper and that of w in [15].) It should be noted that both the required memory size for a user's secret information and that for an authenticated message are increased by this modification. The authors of [15] claimed that their original scheme is optimal in terms of memory sizes for a user's secret information and an authenticated message, however, the fixed version is not. Optimal construction of DMRA in terms of memory sizes for both a user's secret information and an authenticated message is an interesting open problem. We further point out that schemes in [14] and [9] are optimal only for memory sizes for an authenticated message.

3 Two Novel Methods for Constructing Efficient and Unconditionally Secure Digital Signatures

In this section, we show two constructions of unconditionally secure digital signature schemes, which are called *symmetric construction* and *asymmetric construction*, respectively. In these schemes, though the flexibility of parameter settings is partially lost, the required memory sizes are reduced considerably compared to the previous method. More precisely, in our proposed schemes, the number of signatures users can generate is determined to be only one, while in HSZI-AC00 scheme [9], it can be pre-determined flexibly.

3.1 Model

In this subsection, a model of unconditionally secure signature schemes is shown. This model basically follows as in [9] with a restriction of the number of signatures that users can generate.

We assume that there is a trusted authority, denoted by TA, and n users $\mathcal{U} = \{U_1, U_2, \cdots, U_n\}$. For each user $U_i \in \mathcal{U}$ $(1 \leq i \leq n)$, for convenience we use the same symbol U_i to denote the identity of the user. The TA produces secret information on behalf of a user. Once being given the secret information, a user can generate and/or verify signatures by using his own secret information, respectively. A more formal definition is given below:

Definition 1. A scheme Π is an *One-Time Identity-based Signature Scheme for Unconditional Security in a Group (One-Time ISSUSG)* if it is constructed as follows:

1. **Notation:** Π consists of (TA, $\mathcal{U}$, $\mathcal{M}, \mathcal{E}, \mathcal{A}$, **Sig**, **Ver**), where
   - TA is a trusted authority,
   - $\mathcal{U}$ is a finite set of users (to be precise, users' unique names),
   - $\mathcal{M}$ is a finite set of possible messages,
   - $\mathcal{E}$ is a finite set of possible users' secret information,
   - $\mathcal{A}$ is a finite set of possible signatures,
   - **Sig** : $\mathcal{E} \times \mathcal{M} \longrightarrow \mathcal{A}$ is a signing-algorithm,
   - **Ver** : $\mathcal{M} \times \mathcal{A} \times \mathcal{E} \times \mathcal{U} \longrightarrow \{accept, reject\}$ is a verification-algorithm.
2. **Key Pair Generation and Distribution by TA:** For each user $U_i \in \mathcal{U}$, the TA chooses a secret information $e_i \in \mathcal{E}$, and transmits e_i to U_i via a secure channel. After delivering these secret information, the TA may erases e_i from his memory. And each user keeps his secret information secret.
3. **Signature Generation:** For a message $m \in \mathcal{M}$, a user U_i generates a signature $\alpha = \mathbf{Sig}(e_i, m) \in \mathcal{A}$ by using the secret information in conjunction with the signing-algorithm. The pair (m, α) is regarded as a signed message of U_i. After (m, α) is sent by U_i, no user is allowed to generate another signature. Namely, in this scheme only one signature is allowed to be generated, but any user can potentially become a signer.

4. Signature Verification: On receiving (m, α) from U_i, a user U_j checks whether α is valid by using his secret information e_j. More precisely, U_j accepts (m, α) as a valid, signed message from U_i if $\mathbf{Ver}(m, \alpha, e_j, U_i) = \mathit{accept}$.

The main difference between the above definition and the previous one in [9] is that the above model does not allow flexible pre-determination of the number of signatures per user. Hence, this model is called *One-Time* ISSUSG.

For a more formalized discussion for the security of a signature scheme in our model, we define the probability of success of various types of attacks. We consider three broad types of attacks: *impersonation*, *substitution* and *transfer with a trap*. In impersonation, adversaries try to forge a user's signature without seeing the user's valid signature. Note that the adversaries are allowed to observe another user's signature. In substitution, adversaries try to forge a user's signature for a message after seeing the user's valid signature for another message. In transfer with a trap, adversaries try to modify a valid signature to be accepted only by specific verifiers. Description of these attacks are given in [9].

To formally define the probabilities of success in the above three attacks, some notations must be introduced in ahead. Let $\mathcal{W} := \{W \subset \mathcal{U} |\ |W| \leq \omega\}$, where ω is maximum number of colluders among users. Each element of $\mathcal{W}$ represents a group of possibly colluding users. Let $e_W = \{e_{k_1}, \cdots, e_{k_j}\}$, where $W = \{U_{k_1}, \cdots, U_{k_j}\}$ $(j \leq \omega)$, be the set of secret information for a $W \in \mathcal{W}$.

Definition 2. The success probabilities of impersonation, substitution and transfer with a trap attacks, denoted by P_I, P_S and P_T respectively, are formally defined as follows:

1) Success probability of impersonation: for $W \in \mathcal{W}$ and $U_i, U_j \in \mathcal{U}$ with $U_i, U_j \notin W$, we define $P_I(U_i, U_j, W)$ as

$$P_I(U_i, U_j, W) := \max_{e_W} \max_{1 \leq k \leq n, k \neq i} \max_{(m,\alpha)} \max_{(m',\alpha')} \Pr(U_j \text{ accepts } (m', \alpha') \text{ as valid from } U_i | e_W, (m, \alpha)),$$

where (m, α) is a valid signed message generated by a user U_k $(1 \leq k \leq n,\ k \neq i)$ for a message m, and (m, α) runs over $\mathcal{M} \times \mathcal{A}$. Then, P_I is given as $P_I := \max_{\{U_i, U_j, W\}} \Pr(U_i, U_j, W)$, where $W \in \mathcal{W}$ and $U_i, U_j \in \mathcal{U}$ with $U_i, U_j \notin W$.

2) Success probability of substitution: for $W \in \mathcal{W}$ and $U_i, U_j \in \mathcal{U}$ with $U_i, U_j \notin W$, we define $P_S(U_i, U_j, W)$ as

$$P_S(U_i, U_j, W) := \max_{e_W} \max_{(m,\alpha)} \max_{(m',\alpha')} \Pr(U_j \text{ accepts } (m', \alpha') \text{ as valid from } U_i | e_W, (m, \alpha)),$$

where (m, α) is a valid signed message generated by U_i for a message m, and (m', α') runs over $\mathcal{M} \times \mathcal{A}$ such that $m' \neq m$. Then, P_S is given as $P_S := \max_{\{U_i, U_j, W\}} \Pr(U_i, U_j, W)$, where $W \in \mathcal{W}$ and $U_i, U_j \in \mathcal{U}$ with $U_i, U_j \notin W$.

3) Success probability of transfer with a trap: for $W \in \mathcal{W}$ and $U_i, U_j \in \mathcal{U}$ with $U_j \notin W$ we define $P_T(U_i, U_j, W)$ as

$$P_T(U_i, U_j, W) := \max_{e_W} \max_{(m,\alpha)} \max_{(m,\alpha')} \Pr(U_j \text{ accepts } (m, \alpha') \text{ as valid from } U_i | e_W, (m, \alpha)),$$

where (m, α) is a valid signed message generated by U_i, and α' is taken such that $\alpha \neq \alpha'$. Then, P_T is given as $P_T := \max_{\{U_i, U_j, W\}} \Pr(U_i, U_j, W)$, where $W \in \mathcal{W}$ and $U_i, U_j \in \mathcal{U}$ with $U_j \notin W$.

The concept of (n, ω, p_1, p_2)-*secure* One-Time ISSUSG signature scheme can now be defined, where both p_1 and p_2 are security parameters whose meanings will be made precise in the following definition.

Definition 3. Let Π be a One-Time ISSUSG with n users. Then, Π is (n, ω, p_1, p_2)-*secure* if the following conditions are satisfied: as long as there exist at most ω colluders, the following inequalities hold:

$$\max\{P_I,\ P_S\} \leq p_1, \quad P_T \leq p_2.$$

3.2 Symmetric Construction

In this subsection, we show an implementation in One-Time ISSUSG, called the *symmetric construction*. In this construction, the required memory size for a user's secret information is reduced partially based on the fixed version of the SW-DMRA. Namely, we introduce symmetric functions for unifying the secret information for signing and for verifying. However, it should be noted that it is not trivial to implement, since the SW-DMRA does not fulfill the transferability property. The essential reason behind why the SW-DMRA does not provide transferability is that, for U_i's authenticated message $(s_i, a_i(x))$, any entity can calculate $a_i(U_j)$ and find another function $a_i'(x)$ such that $a_i'(x) \neq a_i(x)$ and $a_i'(U_j) = a_i(U_j)$. This is hard to solve since U_j must be public. We show a solution to this problem in the following.

Symmetric Construction

1. Key Generation and Distribution by TA: Let F_{q_0} be the finite field with q_0 elements such that $q_0 \geq n(\omega + 1)q$, where q is a security parameter of the system. We assume that the size of q_0 is almost the same as $n(\omega+1)q$. Then, the TA divides F_{q_0} into n disjoint subsets $\mathcal{U}_1, \cdots, \mathcal{U}_n$, such that $|\mathcal{U}_i| = (\omega+1)q$ for any i, and $\mathcal{U}_i \cap \mathcal{U}_j = \phi$ if $i \neq j$. Here, $\mathcal{U}_i$ $(1 \leq i \leq n)$ are made public for all users. For each user U_i $(1 \leq i \leq n)$, the TA picks uniformly at random, a number u_i from $\mathcal{U}_i$, respectively, and chooses uniformly at random two symmetric polynomials $F_0(x, y), F_1(x, y)$ over F_{q_0} with two variables x and y of degree at most $\omega + 1$. Moreover, we assume a message m is an element in F_{q_0} as well. For each user U_i $(1 \leq i \leq n)$, the TA computes his secret

information $e_i := \{F_0(x, u_i), F_1(x, u_i), u_i\}$. Then, the TA sends e_i to U_i over a secure channel. Once the secret information has been delivered, there is now no need for the TA to keep the user's secret information.

2. **Signature Generation:** For a message $m \in F_{q_0}$, U_i generates a signature by $\alpha := \{a_{i,m}(x), u_i\}$ using his secret information, where $a_{i,m}(x) := F_0(x, u_i) + mF_1(x, u_i)$. Then, (m, α) is sent by U_i with his identity U_i.
3. **Signature Verification:** On receiving U_i's signature (m, α), user U_j checks whether α is valid or not, by the use of his secret information e_j. Specifically, U_j accepts (m, α) as being a valid message-signature pair from U_i if $(F_0(x, u_j) + mF_1(x, u_j))|_{x=u_i} = a_{i,m}(x)|_{x=u_j}$ and $u_i \in \mathcal{U}_i$.

Theorem 1. *The above scheme results in an* $(n, \omega, \frac{1}{q_0}, \frac{1}{q})$*-secure One-Time IS-SUSG scheme.*

Proof: See Appendix.

Theorem 2. *The required memory size in the above construction is given as follows:*

$$|\mathcal{A}| = (\omega + 1)qq_0{}^{\omega+2} \qquad \textit{(size of signature)}$$
$$|\mathcal{E}| = (\omega + 1)qq_0{}^{2\omega+4} \qquad \textit{(size of secret information).}$$

Although in this scheme the required memory size of a signature is slightly increased compared to the HSZI-AC00 scheme [9], that of each user's secret information is significantly reduced. Comparison with the previous method is shown in the following section.

3.3 Asymmetric Construction

In the symmetric construction, though the required memory size of a user's secret information has significantly been reduced, the required memory size of a signature increased compared to the previous method. In this subsection, we show other methods for reducing the required memory size of a user's secret information without increasing the required memory size for a signature. One of the proposed schemes in this subsection is optimal, especially in terms of memory size for a signature. Such schemes are called *asymmetric constructions* since the secret information for signing and that for verification is different.

Asymmetric Construction

1. **Key Pair Generation and Distribution by TA:** Let F_q be the finite field with q elements such that $q \geq n$. The TA picks n elements $v_1, v_2, \ldots, v_n$ uniformly at random in F_q^{ω} for users $U_1, U_2, \ldots, U_n$ respectively, and chooses two polynomials uniformly at random, $G_0(x, y_1, \ldots, y_\omega)$ and $G_1(x, y_1, \ldots, y_\omega)$, over F_q with $\omega + 1$ variables $x, y_1, \cdots, y_\omega$, in which the degree of x is at most $\omega + 1$ and that of every y_i is at most 1. Moreover, we assume that each

user's identity U_i and a message m are elements of F_q. For each user U_i ($1 \leq i \leq n$), the TA computes U_i's secret information $e_i := \{G_0(U_i, y_1, \ldots, y_\omega), G_1(U_i, y_1, \ldots, y_\omega), G_0(x, v_i), G_1(x, v_i), v_i\}$. The TA then sends e_i to U_i over a secure channel. Once all the keys are delivered, there is no need for the TA to keep the user's secret information.

2. **Signature Generation:** For a message $m \in F_q$, U_i generates a signature by $\alpha = G_0(U_i, y_1, \ldots, y_\omega) + mG_1(U_i, y_1, \ldots, y_\omega)$ using $G_0(U_i, y_1, \ldots, y_\omega)$ and $G_1(U_i, y_1, \ldots, y_\omega)$. Then, (m, α) is sent by U_i with his identity U_i.
3. **Signature Verification:** On receiving (m, α) from U_i, user U_j checks whether α is valid by the use of his secret information. More specifically, U_j accepts (m, α) as being a valid message-signature pair from U_i if $(G_0(x, v_i) + mG_1(x, v_i))|_{x=U_i} = \alpha|_{(y_1,\ldots,y_\omega)=(v_{1,j},\ldots,v_{\omega,j})}$.

Theorem 3. *The above scheme results in an* $(n, \omega, (\frac{2}{q} - \frac{1}{q^2}), \frac{1}{q})$*-secure One-Time ISSUSG scheme.*

Similarly to Theorem 1, the proof of Theorem 3 can be given. The above scheme can be slightly modified, resulting in another $(n, \omega, \frac{1}{q}, \frac{1}{q})$-secure One-Time ISSUSG scheme.

Theorem 4. *In the above construction, the following modification also produces an* $(n, \omega, \frac{1}{q}, \frac{1}{q})$*-secure One-Time ISSUSG scheme: Instead of choosing randomly, the TA may choose* n *elements* $v_1, \ldots, v_n \in F_q^\omega$*, for users' secret information, such that for any* $\omega + 1$ *vectors*

$$v_{i_1} = (v_{1,i_1}, \ldots, v_{\omega,i_1}), \ldots, v_{i_{\omega+1}} = (v_{1,i_{\omega+1}}, \ldots, v_{\omega,i_{\omega+1}}),$$

the $\omega + 1$ *new vectors* $(1, v_{1,i_1}, \ldots, v_{\omega,i_1}), \ldots, (1, v_{1,i_{\omega+1}}, \ldots, v_{\omega,i_{\omega+1}})$ *are linearly independent.*

Though the proposed $(n, \omega, \frac{1}{q}, \frac{1}{q})$-secure One-Time ISSUSG scheme is more secure than the proposed $(n, \omega, \frac{2}{q} - \frac{1}{q^2}, \frac{1}{q})$-secure One-Time ISSUSG scheme in terms of impersonation or substitution, it requires more complicated transactions for generating each user's secret information.

Theorem 5. *The required memory size in the above constructions is given as follows:*

$$|\mathcal{A}| = q^{\omega+1} \qquad \textit{(size of signature)}$$
$$|\mathcal{E}| = q^{5\omega+6} \qquad \textit{(size of a user's secret information).}$$

Corollary 1. *The construction proposed in Theorem 4 is optimal in terms of the memory size of a signature.*

The proof follows as from [15]. Since the model of One-Time ISSUSG is regarded as a restricted version of that of MRA, lower bounds on required memory sizes for MRA can also be applied to One-Time ISSUSG. The required memory size for the above construction matches the lower bound on a signature presented in Theorem 5.2 in [15].

Table 1. The required memory sizes of each user's secret information, in the proposed symmetric construction ($(n, \omega, \frac{1}{q_0}, \frac{1}{q})$-secure One-Time ISSUSG), asymmetric construction ($(n, \omega, \frac{1}{q}, \frac{1}{q})$-secure One-Time ISSUSG) and the HSZI-AC00 scheme ($(n, \omega, 1, \frac{1}{q}, \frac{1}{q})$-secure ISSUSG [9]), assuming that $|q| = 160$ bits and ω is determined appropriately for each n.

	$n = 1,000$	$n = 10,000$	$n = 100,000$	$n = 1,000,000$
	$\omega = 500$	$\omega = 2,000$	$\omega = 10,000$	$\omega = 50,000$
Symmetric construction	22Kbyte	91Kbyte	464Kbyte	2,393Kbyte
Asymmetric construction	49Kbyte	196Kbyte	977Kbyte	4,883Kbyte
HSZI-AC00 scheme [9]	69Kbyte	508Kbyte	4,493Kbyte	41,993Kbyte

4 Comparison

In this section, we compare the proposed schemes with the previous method [9]. In the HSZI-AC00 scheme [9], the number of signatures that each user can generate can be pre-determined in a flexible manner. In order to compare the proposed One-Time ISSUSG schemes with the HSZI-AC00 scheme, we set the number of signatures that a user can generate to be one in the previous method. The following proposition shows the required memory sizes for the HSZI-AC00 scheme for this parameter setting.

Proposition 1 ([9]). *Letting the number of users be n and the maximum number of colluders ω, then the required memory sizes for the HSZI-AC00 scheme ($(n, \omega, 1, \frac{1}{q}, \frac{1}{q})$-secure ISSUSG [9][2]) are:*

$$|\mathcal{A}| = q^{\omega+1} \qquad \textit{(size of signature)}$$
$$|\mathcal{E}| = q^{2n+3\omega+2} \qquad \textit{(size of a user's secret information)},$$

assuming that each user is allowed to generate at most 1 signature, the probability of succeeding the impersonation and substitution is at most $\frac{1}{q}$ and that the probability of succeeding transfer with a trap is at most $\frac{1}{q}$.

As shown in the Table 1, the required memory size of each user's secret information is significantly reduced in the proposed schemes. In the symmetric construction, though the required memory size of a signature increases, that of each user's secret information is considerably reduced. As an example, for 100,000 users with appropriate security parameter settings, the required memory size for a user's secret information is reduced to 10.3% of that required in the HSZI-AC00 scheme. In the asymmetric construction, the reduction of

[2] It has now been found that $(n, \omega, \psi, \frac{1}{q}, \frac{1}{q-1})$-secure ISSUSG in [9] is in fact, $(n, \omega, \psi, \frac{1}{q}, \frac{1}{q})$-secure ISSUSG (see the security definition in [9]). Therefore, we have $(n, \omega, 1, \frac{1}{q}, \frac{1}{q-1})$-secure ISSUSG in [9] to be described as $(n, \omega, 1, \frac{1}{q}, \frac{1}{q})$-secure ISSUSG. Details on security of these schemes can be obtained from the present authors.

Table 2. The required memory sizes of a signature, in the proposed symmetric construction ($(n, \omega, \frac{1}{q_0}, \frac{1}{q})$-secure One-Time ISSUSG), asymmetric construction ($(n, \omega, \frac{1}{q}, \frac{1}{q})$-secure One-Time ISSUSG) and the HSZI-AC00 scheme ($(n, \omega, 1, \frac{1}{q}, \frac{1}{q})$-secure ISSUSG [9]), assuming that $|q| = 160$ bits and ω is determined appropriately for each n.

	$n = 1,000$	$n = 10,000$	$n = 100,000$	$n = 1,000,000$
	$\omega = 500$	$\omega = 2,000$	$\omega = 10,000$	$\omega = 50,000$
Symmetric construction	12Kbyte	46Kbyte	233Kbyte	1,197Kbyte
Asymmetric construction	10Kbyte	40Kbyte	196Kbyte	977Kbyte
HSZI-AC00 scheme [9]	10Kbyte	40Kbyte	196Kbyte	977Kbyte

the required memory size of each user's secret information is less than that in the symmetric construction. However, the required memory of a signature is less than that of the symmetric construction. More precisely, the proposed asymmetric construction is optimal in terms of the required memory size of a signature, reminiscent to the HSZI-AC00 scheme. Table 2 shows the required memory sizes for a signature in the proposed schemes and that in the HSZI-AC00 scheme.

References

1. D. Boneh and R. J. Lipton, "Quantum cryptanalysis of hidden linear functions," Proc. of CRYPTO'95, LNCS 963, Springer-Verlag, pp.424-437, 1995.
2. E. F. Brickell and D. R. Stinson, "Authentication codes with multiple arbiters," Proc. of Eurocrypt'88, LNCS 330, Springer-Verlag, pp.51-55, 1988.
3. S. Cavallar, B. Dodson, A. K. Lenstra, et al., "Factorization of a 512-bit RSA modulus," Proc. of Eurocrypt'00, LNCS 1807, Springer-Verlag, pp.1-18, 2000.
4. D. Chaum and S. Roijakkers, "Unconditionally secure digital signatures," Proc. of CRYPTO'90, LNCS 537, Springer-Verlag, pp.206-215, 1990.
5. Y. Desmedt and M. Yung, "Arbitrated unconditionally secure authentication can be unconditionally protected against arbiter's attack," Proc. of CRYPTO'90, LNCS 537, Springer-Verlag, pp.177-188, 1990.
6. Y. Desmedt, Y. Frankel and M. Yung, "Multi-receiver/Multi-sender network security: efficient authenticated multicast/feedback," Proc. of IEEE Infocom'92, pp.2045-2054, 1992.
7. E. N. Gilbert, F. J. MacWilliams and N. J. A. Sloane, "Codes which detect deception," Bell System Technical Journal, 53, pp.405-425, 1974.
8. T. Johansson, "Further results on asymmetric authentication schemes," Information and Computation, 151, pp.100-133, 1999.
9. G. Hanaoka, J. Shikata, Y. Zheng and H. Imai, "Unconditionally secure digital signature schemes admitting transferability," Proc. of Asiacrypt2000, LNCS 1976, Springer-Verlag, pp.130-142, 2000.
10. K. Kurosawa, "New bound on authentication code with arbitration," Proc. of CRYPTO'94, LNCS 839, Springer-Verlag, pp.140-149, 1994.
11. B. Pfitzmann and M. Waidner "Fail-stop signatures and their application," Proc. of Securicom 91, 9th Worldwide Congress on Computer and Communications Security and Protection, pp.145-160, 1991.

12. T. P. Pedersen and B. Pfitzmann, "Fail-stop signatures," SIAM J. on Comp., 26, no.2, pp.291-330, 1997.
13. R. Safavi-Naini and H. Wang, "New results on multi-receiver authentication codes," Proc. of Eurocrypt'98, LNCS 1403, pp.527-541, 1998.
14. R. Safavi-Naini and H. Wang, "Broadcast authentication in group communication," Proc. of Asiacrypt'99, LNCS 1716, Springer-Verlag, pp.399-411, 1999.
15. R. Safavi-Naini and H. Wang, "Multireceiver authentication codes: models, bounds, constructions and extensions," Information and Computation, 151, pp.148-172, 1999.
16. P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," SIAM J. Comp., 26, no.5, pp.1484-1509, 1997.
17. G. J. Simmons, "Authentication theory/coding theory," Proc. of CRYPTO'84, LNCS 196, Springer-Verlag, pp.411-431, 1984.
18. G. J. Simmons, "Message authentication with arbitration of transmitter/receiver disputes," Proc. of Eurocrypt'87, Springer-Verlag, pp.151-165, 1987.
19. G. J. Simmons, "A Cartesian construction for unconditionally secure authentication codes that permit arbitration," Journal of Cryptology, 2, pp.77-104, 1990.

Appendix: Proof of Theorem 1

Assume that after seeing a signed message (m_{i_0}, α) published by U_{i_0}, the colluders $U_1, \cdots, U_\omega$ want to generate (m_{i_1}, α'), such that $m_{i_1} = m_{i_0}$ and the user U_{i_2} will accept it as a valid signed message of the user U_{i_1}, i.e. α consists of $\{u'_{i_1}, a'_{i_1,m_{i_1}}(x)\}$ such that $a'_{i_1,m_{i_1}}(u_{i_2}) = F_0(u'_{i_1}, u_{i_2}) + m_{i_0}F_1(u'_{i_1}, u_{i_2})$ and $u'_{i_1} \in \mathcal{U}_{i_1}$. Letting

$$F_l(x,y) = (1, x, x^2, \cdots, x^{\omega+1}) A_l \begin{pmatrix} 1 \\ y \\ y^2 \\ \vdots \\ y^{\omega+1} \end{pmatrix}, \quad l = 0, 1,$$

where A_l $(l = 0, 1)$ are $(\omega+2)\times(\omega+2)$ symmetric matrices over F_{q_0}, the colluders have a $(\omega + 2) \times (\omega + 1)$ matrix D, where

$$D := (A_0 + m_{i_0} A_1) \begin{pmatrix} 1 & 1 & \cdots & 1 \\ U_{i_0} & U_1 & \cdots & U_\omega \\ {U_{i_0}}^2 & {U_1}^2 & \cdots & {U_\omega}^2 \\ \vdots & \vdots & \cdots & \vdots \\ {U_{i_0}}^{\omega+1} & {U_1}^{\omega+1} & \cdots & {U_\omega}^{\omega+1} \end{pmatrix}.$$

From Lemma 2.1 in [13], there exist q_0 different matrices X such that

$$D = X \begin{pmatrix} 1 & 1 & \cdots & 1 \\ U_{i_0} & U_1 & \cdots & U_\omega \\ {U_{i_0}}^2 & {U_1}^2 & \cdots & {U_\omega}^2 \\ \vdots & \vdots & \cdots & \vdots \\ {U_{i_0}}^{\omega+1} & {U_1}^{\omega+1} & \cdots & {U_\omega}^{\omega+1} \end{pmatrix}.$$

This implies that there are q_0 different values for $A_0 + m_{i_0}A_1$.

In order for the colluders to succeed the attack, they need to find a pair of u'_{i_1} and $a'_{i_1,m_{i_1}}(x)$ such that

$$a'_{i_1,m_{i_1}}(u_{i_2}) = (1, u'_{i_1}, \cdots, u'^{\omega+1}_{i_1})(A_0 + m_{i_0}A_1) \begin{pmatrix} 1 \\ u_{i_2} \\ {u_{i_2}}^2 \\ \vdots \\ {u_{i_2}}^{\omega+1} \end{pmatrix}$$

and $u'_{i_1} \in \mathcal{U}_{i_1}$. Letting d be $(1, u'_{i_1}, \cdots, u'^{\omega+1}_{i_1})(A_0 + m_{i_0}A_1) \begin{pmatrix} 1 \\ u_{i_2} \\ {u_{i_2}}^2 \\ \vdots \\ {u_{i_2}}^{\omega+1} \end{pmatrix}$, q_0 different matrices for $A_0 + m_{i_0}A_1$ result in q_0 different values for d. This indicates that the probability of succeeding to find $a'_{i_1,m_{i_1}}(x)$, such that $a'_{i_1,m_{i_1}}(u_{i_2}) = d$, does not exceed $\frac{1}{q_0}$, i.e. $P_I = \frac{1}{q_0}$. Similarly, we can prove $P_S \leq \frac{1}{q_0}$ and $P_T = \frac{1}{q}$.

Here, we briefly show the proof for $P_T = \frac{1}{q}$. Assume that after seeing a signed message (m_{i_0}, α) published by U_{i_0}, the colluders $U_1, \cdots, U_\omega$ want to generate (m_{i_0}, α'), such that $\alpha' \neq \alpha$ and the user U_{i_1} will accept it as a valid signed message of the user U_{i_0}. Let α be $\{u_{i_0}, a_{i_0,m_{i_0}}(x)\}$ as described in Section 3.2. Since $a_{i_0,m_{i_0}}(x)$ is a polynomial with a variable x of degree at most $\omega + 1$, $a'_{i_0,m_{i_0}}(x)$ $(a'_{i_0,m_{i_0}}(x) \neq a_{i_0,m_{i_0}}(x))$ has at most $\omega + 1$ pairs of $\{c, a'_{i_0,m_{i_0}}(c)\}$, such that $c \in F_{q_0}$ and $a'_{i_0,m_{i_0}}(c) = a_{i_0,m_{i_0}}(c)$, where $a'_{i_0,m_{i_0}}(x)$ is a polynomial with a variable x of degree at most $\omega + 1$. Hence, the best strategy for succeeding transfer with a trap is as follows: The colluders choose uniformly at random $\omega + 1$ distinct numbers $u^{(1)}_{i_1}, \cdots, u^{(\omega+1)}_{i_1}$ from $\mathcal{U}_{i_1}$ and generate $a'_{i_0,m_{i_0}}(x)$ $(a'_{i_0,m_{i_0}}(x) \neq a_{i_0,m_{i_0}}(x))$ such that $a'_{i_0,m_{i_0}}(u^{(1)}_{i_1}) = a_{i_0,m_{i_0}}(u^{(1)}_{i_1})$, $a'_{i_0,m_{i_0}}(u^{(2)}_{i_1}) = a_{i_0,m_{i_0}}(u^{(2)}_{i_1})$, $\cdots$, $a'_{i_0,m_{i_0}}(u^{(\omega+1)}_{i_1}) = a_{i_0,m_{i_0}}(u^{(\omega+1)}_{i_1})$. Then, the colluders send $\alpha' = \{u_{i_0}, a'_{i_0,m_{i_0}}(x)\}$ to U_{i_1}. The attack is successful if and only if $u_{i_1} \in \{u^{(1)}_{i_1}, u^{(2)}_{i_1}, \cdots, u^{(\omega+1)}_{i_1}\}$. Hence, $P_T = \frac{\omega+1}{(\omega+1)q} = \frac{1}{q}$.