Unconditionally Secure Digital Signature Schemes Admitting Transferability

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

[1] The Third Department, Institute of Industrial Science, the University of Tokyo
7-22-1 Roppongi, Minato-ku, Tokyo 106-8558, Japan
Phone & Fax: +81-3-3402-7365
{hanaoka,shikata}@imailab.iis.u-tokyo.ac.jp
imai@iis.u-tokyo.ac.jp
[2] School of Network Computing
Monash University, McMahons Road, Frankston
Melbourne, VIC 3199, Australia
Phone: +61 3 9904 4196, Fax: +61 3 9904 4124
yuliang.zheng@infotech.monash.edu.au

Abstract. A potentially serious problem with current digital signature schemes is that their underlying hard problems from number theory may be solved by an innovative technique or a new generation of computing devices such as quantum computers. Therefore while these signature schemes represent an efficient solution to the short term integrity (unforgeability and non-repudiation) of digital data, they provide no confidence on the long term (say of 20 years) integrity of data signed by these schemes. In this work, we focus on signature schemes whose security does not rely on any unproven assumption. More specifically, we establish a model for unconditionally secure digital signatures in a group, and demonstrate practical schemes in that model. An added advantage of the schemes is that they allow unlimited transfer of signatures without compromising the security of the schemes. Our scheme represents the first unconditionally secure signature that admits provably secure transfer of signatures.

1 Introduction

Digital signatures are an important technology for ensuring the unforgeability and non-repudiation of digital data. While some data may only require the assurance of integrity for a relatively short period of time (say up to 5 years), some other important data, such as court records and speeches by a parliamentarian, require the assurance of integrity for a long period of time (say up to 50 years).

Currently, digital signature schemes based on number theoretic problems are the prevalent methods used in providing data integrity. These schemes rely for their security on the assumed computational difficulty of computing certain number theoretic problems, such as factoring large campsites or solving discrete logarithms in a large finite field. RSA [20], Fiat-Shamir [11], ESIGN [19]

T. Okamoto (Ed.): ASIACRYPT 2000, LNCS 1976, pp. 130–142, 2000.

and many other schemes are based on the difficulty of factoring. On the other hand, ElGamal [10], Schnorr [24], DSA [9] and others, are based on discrete logarithms. Progress in computers as well as further refinement of various algorithms has made it possible to solve the number theoretic problems of larger sizes. As an example, in August 1999, a team of researchers from around the world succeeded in cracking an 512-bit RSA composite by the use of the Number Field Sieve [3] over the Internet. One can safely predict that even larger composites will be factored in the future. In addition, one cannot rule out the possibility of the emergence of innovative algorithms that solve efficiently these number theoretic problems in the future. More importantly, in the past few years there has been significant progress in quantum computers. It has been known that quantum computers can solve both factoring and discrete logarithm problems with ease [25,1], hence advances in the design and manufacturing of quantum computers poses a real threat to the long term security of all the digital signature schemes based on number theoretic problems.

The above discussions indicate the necessity of digital signature schemes that provide assurance of long term integrity. In the past decade, several attempts by various researchers have been made to address the problem. However, schemes proposed by these researchers are essentially variants of authentication codes, and none of these schemes has addressed the transferability of signatures among recipients.

The major contribution of this work is to propose the first digital signature scheme that admits transferability, and provable unconditional security against impersonation, substitution, and transfer with a trap. A potentially useful property of our proposed scheme is that a public key of a user can be associated with the user's unique name, resulting in an identity-based signature scheme.

1.1 Related Work

Chaum and Roijakkers [4] made the first attempt to construct an unconditionally secure signature scheme using cryptographic protocols. Their basic scheme was impractical, as it could only sign a single bit message. Furthermore, in their scheme, the 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. Recently an improved version of Chaum-Roijakkers scheme has been proposed in [14]. However, the author of this improved scheme has not addressed the transferability of his signature scheme.

In another development, Chaum, Heijst and Pfitmann proposed a different version of unconditionally secure signature schemes [5]. However, its unconditional security was guaranteed only for signers.

There have also been attempts to modify unconditionally secure authentication codes [12,26] with the aim of enhancing the codes with extra security properties. It is tempting to transform an unconditionally secure authentication code into a digital signature. There are, however, two technical hurdles that are

hard to overcome. First, authentication codes, especially the conventional Cartesian authentication codes, do not provide the function of non-repudiation, as a receiver can easily forge a sender's message and vice versa. Second, the receiver is always designated, meaning a signature cannot be verified by another party who does not have the shared key. These two properties must be removed for an authentication code to be converted into a digital signature.

An extension of authentication codes is *authentication codes with arbitration* or A^2-codes [27,28,15,16,18,14]. These codes involve a trusted third party called an arbiter. The arbiter can help resolve a dispute when a receiver forges a sender's message or the sender claims that a message is forged by the receiver. A^2-codes have been further improved to require a less trustworthy arbiter. These codes are called A^3-codes [2,7,13,29,14,30]. A property shared by both codes is that the receiver of a signature has to be designated.

As yet another extension, *multi-receiver authentication codes* (MRA) [8,21,14] have been extensively studied in the literature. With a MRA scheme, a broadcast message can be verified by any of the receivers. Although earlier MRA schemes required the sender to be designated, the so-called *MRA with dynamic sender* or DMRA have been proposed [22,23] to relax the requirement of a designated sender. It is important to note that these schemes make sense only in broadcasting. If MRA or DMRA is used for point-to-point authentication, then the sender can easily generate a fraudulent message that is accepted by the receiver, but not by other participants. The situation is made complex due to the fact that the same fraudulent message may have been generated by the receiver himself. A further problem is that MRA or DMRA does not 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 to the next verifier. For the above reasons, neither MRA nor DMRA satisfies the non-repudiation requirement of a digital signature.

In summary, although unconditionally secure authentication codes can be enhanced to satisfy some of the properties of a digital signature, not all of the requirements can be fulfilled. Especially, none of the enhanced authentication schemes had addressed transferability.

1.2 Main Results

In this paper, we present an unconditionally secure identity-based signature scheme. First, we propose a novel model of a signature system called an *Identity-based Signature Schemes for Unconditional Security in a Group* (ISSUSG). As an example implementation of the model, a concrete $(n, \omega, \psi, p_1, p_2)$-*secure* scheme in ISSUSG is demonstrated, where n indicates the total number of users, ω the maximum number of "bad" users who may collude, ψ is the maximum number of signatures a user is allowed to generate, and p_1 and p_2 indicate the best probabilities for an attacker to succeed.

Our approach is an information theoretic one, and the security of our scheme does not rely on any assumption on the computational power of an attacker. Therefore, when the parameters of our scheme are properly chosen, the security

of the scheme will not be affected by future advancement in computing or an algorithmic breakthrough in number theory. An important property of our scheme is that it admits unlimited transfer of signatures from one user to another, without compromising the security of the signature scheme in any way. A further advantage is that the scheme can be made identity-based by associating the unique name of a user to the signature generation and verification algorithms. The scheme is shown to achieve the lower bound on the required memory size of a signature.

As a by-product, we note that our unconditionally secure digital signature scheme can be used as an A^3-code and also as a DMRA. In fact, one may view our scheme as one that fulfills the requirements of both an A^3-code and a DMRA scheme.

The organization of the remaining part of this paper is as follows: In Section 2, we present our new model of an identity-based signature scheme for unconditional security, which we call an *Identity-based Signature Scheme for Unconditionally Security in a Group* (ISSUSG). In Section 3, a concrete unconditionally secure identity-based signature scheme in the model is proposed. In Section 4, some remarks related to our scheme are discussed. Section 5 presents the system-parameter settings when practical memory devices are used. In Section 6, we discuss how to handle long messages in our scheme. Finally, Section 7 concludes the paper with some final remarks.

2 The Model

In the model we consider, signatures are assumed to work in a group. Namely, only members in the group can generate and/or verify signatures. New users are allowed to join the group even after the system is set up, as long as the total number of users does not exceed a pre-defined threshold (this threshold is denoted by n). When the threshold is sufficiently large, in practice our signature scheme can be used in many applications when conventional public key signature schemes are used. Therefore, the group orientation of our scheme should not present any difficulties in practical applications.

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 a pair of signing and verification-keys on behalf of a user. Once being given a pair of keys, a user can then generate and/or verify signatures by using his own signing-key and verification-key, respectively. A more formal definition is given below:

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

1. Notation:

Π consists of (TA, $\mathcal{U}$, $\mathcal{M}, \mathcal{S}, \mathcal{V}, \mathcal{A}$, **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,*
- $\mathcal{S}$ *is a finite set of possible signing-keys,*
- $\mathcal{V}$ *is a finite set of possible verification-keys,*
- $\mathcal{A}$ *is a finite set of possible signatures,*
- $\mathbf{Sig} : \mathcal{S} \times \mathcal{M} \longrightarrow \mathcal{A}$ *is a signing-algorithm,*
- $\mathbf{Ver} : \mathcal{M} \times \mathcal{A} \times \mathcal{V} \times \mathcal{U} \longrightarrow \{accept, reject\}$ *is a verification-algorithm.*

2. Key Pair Generation and Distribution by TA:
For each user $U_i \in \mathcal{U}$*, the TA chooses a signing-key* $s_i \in \mathcal{S}$ *and a verification-key* $v_i \in \mathcal{V}$*, and transmits the pair* (s_i, v_i) *to* U_i *via a secure channel. After delivering these keys, the TA erases the pair* (s_i, v_i) *from his memory. And each user keeps secret both his signing-key and verification-key.*

3. Signature Generation:
For a message $m \in \mathcal{M}$*,* U_i *generates a signature* $\alpha = \mathbf{Sig}(s_i, m) \in \mathcal{A}$ *by using the signing-key in conjunction with the signing-algorithm. The pair* (m, α) *is regarded as a signed message of* U_i*.*

4. Signature Verification:
On receiving (m, α) *from* U_i*, a user* U_j *checks whether* α *is valid by using his verification-key* v_j*. More precisely,* U_j *accepts* (m, α) *as a valid, signed message from* U_i *if and only if* $\mathbf{Ver}(m, \alpha, v_j, U_i) = accept$*.*

The main difference between our definition of signature schemes and that of conventional ones based on public-key cryptography lies in the fact that in our model each user is required to keep secret both his signing-key and verification-key.

In order to discuss in a formal way 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*. Of these attacks, the first two are usually taken into account in discussing the security of authentication codes, especially A^2-codes, A^3-codes, and MRA codes. The third type of attacks, transfer with a trap, is new, and will be formally defined later.

Consider the case where there are n users among whom up to ω user may be dishonest (and hence may collude). Each user is allowed to sign up to ψ signatures. We now discuss in a more formal way the three types of attacks.

1) *Impersonation:*
t users, with $t \leq \omega$, launch an attack against a pair of users U_i and U_j by generating a signed message with the hope that U_j accepts it as being a valid signature from U_i. This attack may be executed after the colluders observe at most $\psi(n-1)$ signed messages generated by users other than U_i.
2) *Substitution:*
t users, with $t \leq \omega$, construct a fraudulent message m' to replace a message genuinely signed by U_i, with the hope that U_j will accept it as being an authentic message from U_i. This attack may be executed after the colluders observe at most ψn signed messages generated by any users. Among the observed messages, at least one but up to ψ may be generated by U_i.

3) *Transfer with a trap:*
After U_j receives a valid pair (m, α) from U_i, t colluders, where $t \leq \omega$, attempt to generate a new pair (m, α') with $\alpha \neq \alpha'$. Note that both the singer U_i and the user U_j could be among the colluders. The colluders hope that another user U_k will accept (m, α') as being a valid message-signature pair from U_i, but no other users will. The risk with this attack is that when U_j transfers such a pair (m, α') to U_k and U_k then transfers it to another user U_l, U_l finds that the pair is invalid. When this happens, U_k is in a sense trapped by the colluders.

To formally define the probabilities of success in the above three attacks, some notations are introduced first.

Let $\mathcal{W} := \{W \subset \mathcal{U} |\ |W| \leq \omega\}$. Each element of $\mathcal{W}$ represents a group of possibly colluding users. Let s_W and v_W be the set of signing-keys and that of verification-keys for a $W \in \mathcal{W}$, respectively.

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

$$P_I(U_i, U_j, W) := \max_{s_W, v_W} \max_{1 \leq k \leq n, k \neq i} \max_{c_k = \{(m_{k,l}, \alpha_{k,l})\}} \max_{(m,\alpha)} \Pr(U_j \text{ accepts } (m, \alpha) \text{ as valid from } U_i | s_W, v_W, \{c_k\}),$$

where $c_k = \{(m_{k,l}, \alpha_{k,l})\}$ is taken over a family of possible sets of valid signed messages generated by U_k $(1 \leq k \leq n,\ k \neq i)$ such that $0 \leq |c_k| \leq \psi$ $(1 \leq k \leq n,\ k \neq i)$. Note that $m_{k,l}$ are not necessarily distinct. 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

$$P_S(U_i, U_j, W) := \max_{s_W, v_W} \max_{1 \leq k \leq n} \max_{c_k = \{(m_{k,l}, \alpha_{k,l})\}} \max_{(m,\alpha)} \Pr(U_j \text{ accepts } (m, \alpha) \text{ as valid from } U_i | s_W, v_W, \{c_k\})$$

where $c_k = \{(m_{k,l}, \alpha_{k,l})\}$ is taken over a family of possible sets of valid signed messages generated by U_k $(1 \leq k \leq n)$ such that $0 < |c_i| \leq \psi$ and $0 \leq |c_k| \leq \psi$ $(1 \leq k \leq n,\ k \neq i)$, and (m, α) is taken such that $m \neq m_{i,l}$ for any l. Note that $m_{k,l}$ are not necessarily distinct. 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*

$$P_T(U_i, U_j, W) := \max_{s_W, v_W} \max_{1 \le k \le n, k \ne i} \max_{c_k = \{(m_{k,l}, \alpha_{k,l})\}} \max_{(m,\alpha)} \max_{(m,\alpha')} \Pr(U_j \text{ accepts } (m, \alpha') \text{ as valid from } U_i | s_W, v_W, \{c_k\}, (m, \alpha))$$

where $c_k = \{(m_{k,l}, \alpha_{k,l})\}$ *is taken over a family of possible sets of valid signed messages generated by* U_k $(1 \le k \le n,\ k \ne i)$ *such that* $0 \le |c_k| \le \psi$ $(1 \le k \le n,\ k \ne i)$, (m, α) *is taken over the set of possible signed messages generated by* U_i, *and* α' *is taken such that* $\alpha \ne \alpha'$. *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* $U_j \notin W$.

Now we are ready to define the concept of an $(n, \omega, \psi, p_1, p_2)$-*secure* ISSUSG signature scheme. Here both p_1 and p_2 are security parameters whose meanings will be made precise in the following definition.

Definition 3 *Let* Π *be an ISSUSG with* n *users. Then,* Π *is* $(n, \omega, \psi, p_1, p_2)$-*secure if the following conditions are satisfied: as long as there exist at most* ω *colluders and each user is allowed to generate at most* ψ *signatures, the following inequalities hold:*

$$\max\{P_I,\ P_S\} \le p_1$$
$$P_T \le p_2$$

where P_I, P_S *and* P_T *are the probabilities of success in impersonation, substitution and transfer with a trap attacks, respectively.*

We note that there is an alternative definition of security in which one may use a single security parameter p instead and define the success probability as

$$\max\{P_I,\ P_S,\ P_T\} \le p.$$

In practice, however, some applications may attach more weight to strength against impersonation and substitution than against transfer with a trap, while some other applications may have an emphasis on robustness against transfer with a trap. By introducing two separate parameters p_1 and p_2, we have an opportunity to design a signature scheme with fine-tuned level of security.

3 Implementation

3.1 Protocol

In this section, an implementation of the ISSUSG will be presented. It is constructed by the use of a polynomial with $\omega + 2$ variables over a finite field.

As before, let $\mathcal{U} := \{U_1, U_2, \cdots, U_n\}$ be the set of n users and TA the trusted authority.

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 uniformly at random n elements $v_1, v_2, \ldots, v_n$ in F_q^ω for users $U_1, U_2, \ldots, U_n$ respectively, and constructs a polynomial $F(x, y_1, \ldots, y_\omega, z)$ as follows:

$$F(x, y_1, \ldots, y_\omega, z) = \sum_{i=0}^{n-1} \sum_{k=0}^{\psi} a_{i0k} x^i z^k + \sum_{i=0}^{n-1} \sum_{j=1}^{\omega} \sum_{k=0}^{\psi} a_{ijk} x^i y_j z^k$$

where the coefficients a_{ijk} are chosen uniformally at random from F_q. Moreover, we assume that a user's identity U_l and a message m are also from F_q.

For each user U_l $(1 \leq l \leq n)$, the TA computes a *signing-key* $s_l := F(U_l, y_1, \ldots, y_\omega, z)$, and a *verification-key* $\tilde{v_l} := F(x, v_l, z)$. v_l and $\tilde{v_l}$ together form a pair of verification-keys for user U_l. The TA then sends both the signing-key and the pair of verification-keys to U_l over a secure channel. Once the keys are delivered, there is no need for the TA to keep the user's keys.

2. Signature Generation:

For a message $m \in F_q$, U_i generates a signature by

$$\alpha = F(U_i, y_1, \ldots, y_\omega, z)|_{z=m} = F(U_i, y_1, \ldots, y_\omega, m)$$

using his signing-key.

3. Signature Verification:

On receiving (m, α) from U_i, user U_j checks whether α is valid by the use of his verification-keys v_j and $\tilde{v_j}$. More specifically, U_j calculates evaluation values r_1, r_2 using his verification-keys $\tilde{v_j} = F(x, v_j, z)$ and $v_j := (v_{1,j}, \ldots, v_{\omega,j})$ as follows:

$$r_1 := F(x, v_j, z)|_{x=U_i, z=m},$$
$$r_2 := \alpha|_{(y_1,\ldots,y_\omega)=(v_{1,j},\ldots,v_{\omega,j})}.$$

U_j accepts (m, α) as being a valid message-signature pair from U_i if and only if $r_1 = r_2$.

We can show that the above signature scheme is an $(n, \omega, \psi, (\frac{2}{q} - \frac{1}{q^2}), \frac{1}{q})$-secure ISSUSG scheme.

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

Due to the lack of space, the proof of Theorem 1 is omitted. It will be provided in the full version of this paper.

The above scheme can be modified slightly, resulting in yet another $(n, \omega, \psi, \frac{1}{q}, \frac{1}{q-1})$-secure ISSUSG scheme.

Theorem 2 *In the above construction, the following modification produces also an* $(n, \omega, \psi, \frac{1}{q}, \frac{1}{q-1})$*-secure ISSUSG scheme:*
Instead of choosing randomly, the TA may choose n *elements* $v_1, \ldots, v_n \in F_q^{\omega}$*, as verification-keys, 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.*

Note that our scheme can be used in place of an authentication code, MRA or DMRA. In fact our scheme is cryptographically stronger than the authentication codes, with an added benefit of being transferable, although it requires more memory space than MRA and DMRA.

3.2 Memory Sizes

The following theorem states the required memory size for our construction, and its proof is obvious.

Theorem 3 *The required memory size in the above constructions is given as follows:*

$$\begin{aligned} |\mathcal{A}| &= q^{(\omega+1)}, && \textit{(size of signature)} \\ |\mathcal{S}| &= q^{(\omega+1)(\psi+1)}, && \textit{(size of signing-key)} \\ |\mathcal{V}| &= q^{\omega+n(\psi+1)}, && \textit{(size of verification-key).} \end{aligned}$$

Corollary 1 *The construction proposed in Theorem 2 is optimal in terms of the memory size of a signature.*

The proof follows from [23].

It is not yet clear to the authors as to whether the scheme also achieves optimality in terms of memory size for signing-keys and verification-keys.

4 Some Remarks on Our Scheme

This section shows useful extensions of the scheme presented above, and discusses some of the properties of the scheme. More detailed discussions will be provided in the full version of this paper.

4.1 Signature Scheme for t Senders

In some applications, users who might sign are specified first. When there are only t specified senders in the system, we can easily specialize our scheme to produce a *signature scheme for* t *senders*. Namely, by changing the degree $n-1$ of x in $F(x, y_1, \cdots, y_\omega, z)$ to $t-1$, a signature scheme for t senders is obtained. Based on this restriction, the required memory for verification-keys can be reduced from $q^{\omega+n(\psi+1)}$ to $q^{\omega+t(\psi+1)}$. Note that the required memory sizes for signatures and signing-keys are still the same as in the non-restrictive scheme.

4.2 Arbiter

We can also introduce an *arbiter* which can resolve a dispute between a signer and a recipient. In one such implementation, the arbiter will be given a pair of verification-keys, whereas no user will. The arbiter can notify users of the result of verification of a signature. We note that any user can play the role of an arbiter for other users.

4.3 Reduction of Memory Size for Verification-Key

In the proposed schemes in Section 3, the degree of x in $F(x, y_1, \cdots, y_\omega, z)$ is set as $n-1$. If the degree of x is $\omega + d$ instead ($d \leq n - \omega - 2$), the system may be attacked as follows: when the same message is signed by $d+1$ signers, ω colluders can forge a victim's signature of the same message by using their own secrets and the generated signatures. To prevent the scheme from this attack, the degree of x is set to $n-1$, which is the primary contributor to the required memory size of verification-keys.

If in a practical system it is known that the chance for the same message to be signed by $d+1$ signers is extremely small, the degree of x may be set to be smaller than $n-1$. This will reduce the required memory size for verification-keys.

4.4 Active Attacks against Verification-Keys

As already discussed earlier, the proposed scheme is unconditionally secure against passive attacks. In an active attack, an adversary may manage to obtain some information on verification-keys. As an example, by selecting a random element from $F_q^{\omega+1}$ as a forged signature and obtaining the verification result from a targeted victim, the adversary obtains some information on the victim's verification-key. We can show that the information obtained does not help succeed with a non-negligible probability in impersonation, substitution or transfer with a trap. Thus such an active attack is not an issue in practice. Details will be presented in the full paper.

5 Practical Systems Based on Memory Devices

In this section, we discuss the values of security parameters in the proposed schemes. Table 1 shows the value of ψ according to the values of the number of users and memory devices which may contain users' signing-keys, assuming the worst case where $\omega = n-1$. One can see that using commonly available memory devices, the number of signatures that can be generated by a user is sufficiently large even for a large organization that has 1,000 to 10,000 users.

Table 2 gives data on a more realistic setting. One can see that compared with the previous table, the number of signatures that can be signed by a user increases significantly.

We note that the capacity of memory devices is getting larger and larger, and their prices are dropping as fast. This helps significantly the usability of the proposed signature scheme.

Table 1. The number of signatures a user can generate, assuming that $|q|$ has 160 bits and $\omega = n - 1$.

	$n = 1,000$	$n = 10,000$	$n = 100,000$	$n = 1,000,000$
2HD disk(1.44MByte)	71	6	0	0
ZIP(100MByte)	4,999	499	49	4
CD-R(650MByte)	32,499	3,249	324	31
DVD-RAM(5.2GByte)	259,999	25,999	2,599	259

Table 2. The number of signatures a user can generate, assuming that $|q|$ has 160 bits and ω is determined appropriately for each n.

	$n = 1,000$	$n = 10,000$	$n = 100,000$	$n = 1,000,000$
	$\omega = 500$	$\omega = 2,000$	$\omega = 10,000$	$\omega = 50,000$
2HD disk(1.44MByte)	142	34	6	0
ZIP(100MByte)	9,979	2,497	498	98
CD-R(650MByte)	64,869	16,240	3,248	648
DVD-RAM(5.2GByte)	518,961	129,934	25,996	5,198

6 On Handling Long Messages

In our proposed scheme, the length of messages to be signed is restricted to be $|q|$ or less. An important question that is yet to be addressed is how to handle longer messages, without significantly increasing the size of such a message.

In practice, one may use the technique of applying a one-way hashing to a long message prior to signing it. Some examples of one-way hash algorithms are SHA-1 [17], HAVAL [31] and RIPEMD-160 [6]. Although this will lose the unconditional security property of the proposed signature scheme, we note that a good one-way hash function would remain secure even if one employed quantum computers in attacking it.

7 Conclusions

We have proposed unconditionally secure identity-based signature schemes. More specifically, we have established a model for unconditionally secure digital signatures in a group, and demonstrated practical schemes in that model. An added advantage of the scheme is that it allows unlimited transfer of signatures without compromising the security of the scheme. Although there is a limit on the number of signatures a user can generate, this limitation is not an issue in practice thanks to the development in inexpensive memory devices with a huge capacity. Specifically, by using a DVD-RAM, 25,999 signatures can be generated by a user in an organization of 10,000 employees.

We are currently working on other possible implementations of ISSUSG, as well as the problem on how to sign long message without losing unconditional security.

Acknowledgments

Part of this work was supported by Research for the Future Program (RFTF), Japan Society for the Promotion of Science (JSPS), under contract number JSPS-RETF 96P00604. The first author was supported by a Research Fellowship from JSPS.

The authors would like to thank Tsutomu Matsumoto and Tsuyoshi Nishioka for their valuable comments. The first author would like also to thank Yumiko C. Hanaoka for her help in preparing this paper.

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. D. Chaum, E. Heijst and B. Pfitzmann, "Cryptographically strong undeniable signatures, unconditionally secure for the signer," Proc. of CRYPTO'91, LNCS 576, Springer-Verlag, pp.470-484, 1991.
6. H. Dobbertin, A. Bosselaers and B. Preneel, "RIPEMD160: strengthened version of RIPEMD," Proc. of FSE'96, LNCS 1039, Springer-Verlag, pp.71-82, 1996.
7. 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.
8. 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.
9. "Proposed federal information processing standard for digital signature standard (DSS)," Federal Register, vol. 56, no. 169, 30, pp.42980-42982, 1991.
10. T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," IEEE Trans. on Inform. Theory, IT-31, 4, pp.469-472, 1985.
11. A. Fiat and A. Shamir, "How to prove yourself: practical solutions to identification and signature problems," Proc. of CRYPTO'86, LNCS 263, Springer-Verlag, pp.186-194, 1986.
12. E. N. Gilbert, F. J. MacWilliams and N. J. A. Sloane, "Codes which detect deception," Bell System Technical Journal, 53, pp.405-425, 1974.
13. T. Johansson, "Lower bounds on the probability of deception in authentication with arbitration", IEEE Trans. Inform. Theory, IT-40, 5, pp.1573-1585, 1994.
14. T. Johansson, "Further results on asymmetric authentication schemes," Information and Computation, 151, pp.100-133, 1999.

15. K. Kurosawa, "New bound on authentication code with arbitration," Proc. of CRYPTO'94, LNCS 839, Springer-Verlag, pp.140-149, 1994.
16. K. Kurosawa and S. Obana, "Combinatorial bounds for authentication codes with arbitration," Proc. of Eurocrypt'95, LNCS 921, Springer-Verlag, pp.289-300, 1995.
17. NIST, "Secure hash standard," *FIPS PUB 180-1*, Department of Commerce, Washington D.C., 1995.
18. S. Obana and K. Kurosawa, "A^2-code = affine resolvable + BIBD," Proc. of ICICS'97, LNCS 1334, Springer-Verlag, pp.118-129, 1997.
19. T. Okamoto, "A fast signature scheme based on congruential polynomial operations," IEEE Trans. on Inform. Theory, IT-36, 1, pp.47-53, 1990.
20. R. Rivest, A. Shamir and L. Adleman, "A method for obtaining digital signature and public-key cryptosystems," Communication of the ACM, vol.21, no.2, pp.120-126, 1978.
21. R. Safavi-Naini and H. Wang, "New results on multi-receiver authentication codes," Proc. of Eurocrypt'98, LNCS1403, pp.527-541, 1998.
22. R. Safavi-Naini and H. Wang, "Broadcast authentication in group communication," Proc. of Asiacrypt'99, LNCS1716, Springer-Verlag, pp.399-411, 1999.
23. R. Safavi-Naini and H. Wang, "Multireceiver authentication codes: models, bounds, constructions and extensions," Information and Computation, 151, pp.148-172, 1999.
24. C. Schnorr, "Efficient signature generation by smart cards," Journal of Cryptology, 4, pp.161-174, 1991.
25. 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.
26. G. J. Simmons, "Authentication theory/coding theory," Proc. of CRYPTO'84, LNCS 196, Springer-Verlag, pp.411-431, 1984.
27. G. J. Simmons, "Message authentication with arbitration of transmitter/ receiver disputes," Proc. of Eurocyrpt'87, Springer-Verlag, pp.151-165, 1987.
28. G. J. Simmons, "A Cartesian construction for unconditionally secure authentication codes that permit arbitration," Journal of Cryptology, 2, pp.77-104, 1990.
29. R. Taylor, "Near optimal unconditionally secure authentication," Proc. of Eurocyrpt'94, LNCS 950, Springer-Verlag, pp.244-253, 1994.
30. Y. Wang and R. Safavi-Naini, "A^3-codes under collusion attacks," Proc. of Asiacrypt'99, LNCS 1716, Springer-Verlag, pp.390-398, 1999.
31. Y. Zheng, J. Pieprzyk and J. Seberry, "HAVAL - A one-way hashing algorithm with variable length of output," Proc. of Auscrypt'92, LNCS 718, Springer-Verlag, pp.83-104, 1993.