*Regular Paper*

# Design of Self-Delegation for Mobile Terminals

Shinsaku Kiyomoto,[†] Toshiaki Tanaka,[†] Mariko Yoshida[††]
and Masahiro Kuroda[†††]

In this paper, we propose a new authentication mechanism for the mobile environments, called *Self-Delegation*. In the mechanism, a user stores information that relates to strict authentication in a tamper-resistant module that can be kept securely at home. Time-limited authority is delegated to the mobile terminal by communicating with the tamper-resistant module on a local basis. After the delegation, a remote service can authenticate the user for a limited time. We propose two self-delegation schemes, and analyze the security of the proposed scheme based on a security model that we define. Furthermore, we have implemented the self-delegation and authentication protocols on a PDA and a Java card, both of which have ISO14443 I/F, and show the feasibility of the implemented protocols.

## 1. Introduction

The 3G cellular system has infiltrated the current market, and the next generation wireless system beyond 3G will be set to gain market share. The Beyond 3G system integrates various wireless accesses, including 3G, 4G, and wireless LANs. It provides an all-IP wireless solution to offer services, taking advantage of each wireless communication mechanism in the system. There is active research into integrating heterogeneous wireless networks into an all-IP network using IP technologies. An architectures discussed at the new generation mobile communication project[1)] is a potential solution for the Beyond3G system. For the next generation mobile services, personal authentication is needed to provide value-added and fine-grained services such as mobile commerce, mobile banking, etc.

In this paper, we propose a mechanism by which strict authentication may be realized even if the off-the-shelf mobile terminal has no tamper-resistant module. A user stores information that relates to strict authentication in a tamper-resistant module, which can be stored securely at home. Time-limited authority is then delegated to a mobile terminal by communicating with the tamper-resistant module on a local basis. After the delegation, a remote service can authenticate the user for a limited time.

The basic concept is called self-delegation. Goldreich proposed a self-delegation method with controlled propagation based on a non-interactive zero-knowledge proof in Ref. 2). Other approaches to the issue of limiting the delegation of rights were suggested in Ref. 3). These schemes, however, do not apply to mobile environments, where one must consider computational cost. We propose a practical approach for self-delegation in mobile environments. The proposed mechanism uses both primary and secondary key information. First, the primary key information is stored in the tamper-resistant module. During the self-delegation, secondary key information is derived from the primary key information by the tamper-resistant module and is securely installed in the mobile terminal. The user can then be authenticated using the secondary key in the mobile terminal.

The proposed scheme is a solution for an open problem in general authentication schemes, how to authenticate users who have lost or have had their key information stolen. If the mobile terminal is stolen, a service provider can authenticate the user using the primary key information stored in the tamper-resistant module. After authentication, the user can request that the service provider revoke the secondary key.

A Kerberos-based scheme such as the scheme proposed by Fox, et al.[16)], is a feasible solution for indirect authentication in mobile environments. However, direct and real-time authentication is required by sensitive services such as e-commerce. The self-delegation proposed here provides direct and strict authentication securely. Self-delegation is widely applicable to any services that authenticates a user, such as

† KDDI R & D Laboratories Inc.
†† Mitsubishi Electric Corporation
††† National Institute of Information and Communications Technology

e-commerce, identification, and others.

The rest of the paper is organized as follows: first, we present the concept of self-delegation in Section 2; then we propose two self-delegation schemes in Section 3. The security analysis of the proposed schemes is discussed in Section 4. Section 5 presents evaluation results. Applications using self-delegation are introduced in Section 6. Finally, we conclude the paper in Section 7.

## 2. Concept of Self-Delegation

In current mobile services, the mobile operator authenticates to a subscriber by using identification stored in a mobile terminal; the identification is assigned when the user subscribes. In such case, the mobile operator does not identify the person, but rather the user's mobile terminal. For next generation mobile services, however, personal identification is needed to provide value-added and fine-grained services such as mobile commerce, mobile banking, etc. One solution is to store some personal identification mechanisms such as biometrics in the mobile terminal to activate it. However, this solution has the following drawbacks.

- Sensitive information or mechanisms for personal identification cannot be securely managed on a mobile terminal because a mobile terminal may be easily stolen or lost. To avoid this security risk, the mobile terminal requires an additional secure element such as an IC chip, which is not cost effective.
- Biometrics mechanisms also require additional user action for authentication. For example, the user has to place their finger on the CCD sensor to allow their fingerprint to be scanned. If the mobile operator requires strict authentication, the user must perform this action to access to each service.

To solve such problems, we apply a mechanism by which the user is strictly authenticated even if the off-the-shelf mobile terminal has no tamper-resistant module. The basic idea of self-delegation is as follows; a user stores information that relates to strict authentication in a tamper-resistant module, which can be stored securely at home. Time-limited authority is delegated to the mobile terminal by communicating with the tamper-resistant module on a local basis. After delegation, the user can be authenticated to remote services for a limited time via the mobile terminal.

Some schemes that are usable for a self-delegation scheme have been proposed such as Goldreich schemes and Chaum's scheme. However, their applicability to mobile environments is not discussed. Thus, optimizations and modifications of the schemes will be mandatory, when the schemes are applied for mobile environments. We designed self-delegation schemes applicable to the mobile environments and evaluete them. The proposed schemes use a hierarchy of two keys, and delegate a secondary key that is verifiable using the primary key. First, the primary key information is stored in the tamper-resistant module. During self-delegation, the secondary key information is derived from the primary key information by the tamper-resistant module, and is securely installed in the mobile terminal. The tamper-resistant module also stores a part of the secondary key information. The user can be authenticated by using the secondary key in the mobile terminal. The secondary key has limited validity, and if expired, the information is of no use for authentication. The service provider checks the term of validity after authentication, and if expired, they do not allow the service to be used. The mobile terminal also checks the term of validity, so that expired information is automatically deleted. Even if an attacker obtains a mobile terminal or delegated information, they are only able to use it for a limited time.

If the mobile terminal is stolen, a service provider can also authenticate the user using the primary key information stored in the tamper-resistant module. The user may then send the information of the stolen secondary key, which is also stored into the tamper-resistant module, and request that the service provider revokes the secondary key. After the revocation, the service provider adds identifier of the stolen key to a revocation list. A stolen mobile terminal having the key cannot use the service.

Self-delegation is widely applicable to services that strictly authenticate a user, such as e-commerce, identification, and other popular services. Currently, a user tends to have many IC cards that are issued by different service providers. Using self-delegation, a user does not need to carry all their IC cards but instead only has to carry their mobile terminal, which has delegated information from the cards.

## 3. Self-Delegation between a Mobile Terminal and Smart Cards

### 3.1 Preparation

We assume that the primary key is stored in a tamper-resistant module called a Personal Identification Card (PIC)[20]. A PIC has a public/private key pair that identifies the holder. First, a user registers with service providers. Service providers provide secret information to the user. The secret information is stored in a PIC, and will be delegated to the user's mobile terminal. We describe a mobile terminal and a PIC as follows;

- Mobile Terminal (MT)
  The Mobile Terminal is used as an user terminal in mobile environments, such as cellular phone and PDA. Delegated information can be stored in the mobile terminal. Mobile terminals have an interface module to communicate securely with PICs.
- Personal Identification Card (PIC)
  A PIC has a processor and power supply. A PIC can compute by itself. A PIC is trusted by the holder, and communication between a mobile terminal and a PIC is protected using current technologies. A PIC can be used for authentication and/or identification of the holder, and stores information for authentication. A PIC has a strict holder-identification mechanism, and identifies the holder when the PIC is used. A user usually does not carry the PIC, and the PIC is stored in a secure location such as the user's home.

A user delegates authentication information from primary key information to the mobile terminal. Whenusing a service, the mobile terminal is authenticated via the delegated information, without the PIC, through a service authentication protocol.

Registration with service providers is required to use services. In the registration phase, a service provider first authenticates a user, and then distributes primary information to the user via a secure channel. The primary information is a secret key or a public key certificate, which is issued by the service provider. Many currently available techniques apply to the service registration.

### 3.2 Requirements

Below, we describe basic requirements for a self-delegation and service authentication protocol, which can be applied to mobile terminals and personal identification card.

- Computational Cost
  The mobile terminal and the personal identification module have low computational power. Therefore, the computational cost of the protocols should be small.
- Symmetric and Asymmetric
  There are two main types of authentication: schemes using an asymmetric key algorithm, and schemes using a symmetric key algorithm. To have self-delegation apply to many various services, both schemes should be considered. The symmetric-key-based scheme is generally faster than the public-key-based scheme, even though the key management cost is not negligible. However, it is only applicable for a single service provider. If we assume an environment in which each service provider manages user information, such as account information, then we will want a symmetric-key-based scheme. The public-key-based schemes are applicable for multi-domain environments. Thus, the scheme can provide an environment in which one service provider provides a user information management and accounting service to other service providers.
- Local Delegation
  This is the most basic requirement for the self-delegation. Users should be able to perform self-delegation themselves without a trusted third party. Furthermore, self-delegation should only use local communication, not network access.

Consider a simple delegation scheme in which a service provider engages in a self-delegation. The service provider authenticates the user based on the primary information and issues the secondary information for delegation. In this example scheme, the service provider authenticates only using the delegated information, because the service provider knows the information, but the scheme requires access to global networks for the delegation. On the other hand, we construct our self-delegation scheme to have the feature that self-delegation can be executed only with only local communication between the MT and the PIC.

A key point is the relation between primary information and delegated information. The service provider can verify the validity of primary information. Thus, if the service provider can verify or compute the delegated information using the primary information, the authentication protocol based on the delegated information can be constructed. Therefore, our design policy is to make a secure relation between the primary and the secondary key that is verifiable by the service provider. Furthermore, the self-delegation should be a lightweight protocol and the computational cost should be sufficiently small.

Next, we discuss security requirements considering the above discussion. Security requirements for our proposed self-delegation are as follows;

- A delegated key is not independent of the primary key. A service provider can verify that the delegated key is computed from the primary key.
- A primary key cannot be computed from delegated keys.
- A delegated key is a time-limited key. An adversary cannot alter the term of validity of a delegated key.
- A delegated key is computationally secure against forgery. An adversary cannot forge a delegated key.

### 3.3 Design Strategy of Self-delegation

This subsection describes our main strategy for designing self-delegation schemes. We design a simple self-delegation protocol for symmetric encryption using a fast one-way function such as a Message Authentication Code (MAC) algorithm. Thus, the scheme requires a small computation and the size of the delegated information, including the delegated key, is also small. In the scheme, the delegated key is computed not only from a random number and the primary key, but also related information including the term of validity. An adversary cannot alter the term of validity without the primary key. A service provider can compute the secondary key from the primary key and the random number in the same manner that the PIC does during self-delegation. Thus, the scheme achieves a local delegation.

On the other hand, to apply public-key-based authentication, we designed a ticket-based self-delegation. Although a simple solution would be to delegate a new public/private key pair that is signed using a primary private key, we designed the ticket-based scheme also considering the computational cost in authentication phase. The ticket reduces the computational cost of authentication protocols, because the MT's key for authentication is transformed from an asymmetric key to a symmetric key by the self-delegation, and the symmetric key is encapsulated into the ticket which is verifiable using the primary pubic key. The total cost of authentication on a mobile terminal that is authenticated many times can be reduced dramatically, even though one public key encryption on a PIC is required in the self-delegation. Furthermore, the cost of generating a new public/private key pair on the PIC is eliminated. This property reduces computational cost for the PIC, especially in the case of RSA-based public key algorithms.

The data to sign digitally includes the term of validity. Thus, the term of validity is unforgeable. The random seed for the secondary key is encrypted using the public key of the service provider and is stored into the ticket. Then, the service provider can computes the secondary key from the seed and some public information. Thus, this scheme also achieves a local delegation.

We present details of the self-delegation in the next subsection, and discuss the security of the self-delegation in Section 4.

### 3.4 Self-Delegation and Authentication Protocols

The PIC transfers delegated information to the MT. We designed two pairs of delegation and authentication protocols. The delegation is selected according to primary information stored in the PIC. If primary information is a secret key, a secondary secret key is delegated. If primary information is a public/private key pair, a new digitally signed ticket and a secret key is delegated.

*Notation.* Let $Ks_x$, $Kpu_x$, $Kpr_x$, $Uid$, $Info$, be the primary secret key of $x$, the primary public key of $x$ including the public key's certificate, the primary private key, a user ID, and additional information such as validity, respectively. $Km_x$ is a master secret key of $x$ that only $x$ knows. $K_{auth}$ indicates a delegated secondary key, and $Ticket$ indicates a delegated ticket. $||$ means data concatenation, and message authentication code of $x$ by key $k$ is described $M(k,x)$, and $M(k,x)$. $X \rightarrow Y$ indicates data communication from $X$ to $Y$. $E(k,x)$, and D(k, x) mean public key encryption of $x$ using a key $k$, and decryption of $x$ using a key $k$, respectively. $S(k,x)$ means a digital signature of $x$ by a private key $k$. $R$, $R'$, $R''$ are random numbers. $U$ is a user, and $SP$ indicates the service provider that issues the user's primary key.

Self-delegation schemes are as follows:

**Secret-key-based Self-Delegation**
$PIC : K_{auth} = M(Ks_U, Uid||R||Info)$
$PIC \rightarrow MT : K_{auth}, Uid, Info, R$

**Ticket-based Self-Delegation**
$PIC : Ticket =$
$Uid||Info||E(Kpu_{SP}, R)||S(Kpr_U, Uid$
$||Info||E(Kpu_{SP}, R))$
$PIC : K_{auth} = M(R, Uid)$
$PIC \rightarrow MT : K_{auth}, Ticket$

The following authentication protocols use the delegated secret key and ticket. In the authentication protocols, we assume each entity identifies the other using out-of-band data such as a network address. We describe the authentication protocols focusing on authentication mechanisms, without the unrelated data.

**Secret-key-based Auth. Protocol**
$SP \rightarrow MT : R'$
$MT \rightarrow SP : Uid, R, R'', Info,$
$M(K_{auth}, Uid||R||R'||R''||Info)$
$SP : M(Km_{SP}, Uid) = Ks_U$
$SP : M(Ks_U, Uid||R||Info) = K_{auth}$
The $SP$ verifies
$M(K_{auth}, Uid||R||R'||R''||Info)$ from $MT$.
$SP \rightarrow MT : M(K_{auth}, R'')$
The $MT$ verifies $M(K_{auth}, R'')$ from $SP$.

**Ticket-based Auth. Protocol**
$SP \rightarrow MT : R'$
$MT \rightarrow SP : Uid, R'', Info, Ticket, Kpu_U, M($
$K_{auth}, Uid||R'||R''||Info||Ticket||Kpu_U)$
The $SP$ verifies $KpuU$ and the signature of the ticket.
$SP : D(Kpr_{SP}, E(Kpu_{SP}, R)) = R$
$SP : M(R, Uid) = K_{auth}$
The $SP$ verifies $M(K_{auth},$
$Uid||R'||R''||Info||Ticket||Kpu_U)$
$SP \rightarrow MT : M(K_{auth}, R'')$
The $MT$ verifies $M(K_{auth}, R'')$ from $SP$.

## 4. Security Analysis

In this section, we analyze the security of our proposed schemes. In self-delegation, users of mobile services just have to carry their mobile terminal, and the tamper-resistant module can be stored in a secure location such as their homes, because, an authentication using self-delegation only requires the secondary information. Thus, the security of the primary information will be improved. The delegated information can be used only for a limited time. Thus, if an attacker steals the mobile terminal, they can only use the services for the limited time. Furthermore, a revocation scheme of delegated information is established. The user can securely request that the revoke the service provider revoke the delegated information, because the service provider can authenticate the user based on the primary information.

We assume that the tamper-resistant module has a strict identification mechanism for the user. Even if the module is stolen by an attacker, the attacker cannot use the module. Thus, the consequence of an attacker obtaining the tamper-resistant module, or the module and the mobile terminal, are almost same as the case where only the mobile terminal is stolen.

Now, we discuss a cryptographic analysis of the proposed schemes. We assume that an adversary has correct secondary keys stored into a mobile terminal such as unauthorized borrowing. The adversary can uses a mobile service within the term of validity of the stolen secondary key, and this is a security specification of self-delegation. However, if an adversary can generate a new, correct secondary key, a service provider allows the adversary to use the service beyond the limited term. Thus, we have to analyze whether an adversary can make a new secondary key in proposed scheme. To analyze the security, we discuss the unforgeability of the secondary key, i.e., whether an adversary can make a new secondary key without the help of the PIC. The condition of being indistinguishable from random strings indicates that the adversary cannot check whether a forged key is correct.

We also discuss the *Forward Secrecy* of the proposed self-delegation. This model assumes the adversary can obtain a primary key using some special techniques without breaking the tamper-resistance of the PIC. It is desirable to satisfy this security notion where the secondary key is used for a key exchange protocol. For example, if a self-delegation scheme has *Forward Secrecy* and a key exchange protocol is secure under the assumption that the secondary key is secure, then a session key is secure. Thus, the security of previous communications is guaranteed on the condition that the self-delegation has *Forward Secrecy*, even if the primary key is compromised.

An adversarial model of *Forward Secrecy* may be an unrealistic model, because a primary key is usually stored in the PIC and we assume that the PIC is stored in a secure location. However, we believe this analysis is useful for clarifying the difference between secret-key-based scheme and public-key-based scheme.

Furthermore, we analyze the security of authentication protocols. In this discussion, we consider the provable security of the authentication protocols as proposed by Bellare and Rogaway. An adversarial model of this security notion implicitly includes several attacks against authentication protocols such as forgery attacks and replay attacks.

## 4.1 Security Analysis of Self-Delegation

We discuss the security of the self-delegation schemes in this subsection. First, we define secure self-delegation as follows.

**Definition 1.** We assume an adversary who can obtain any secondary information without a $K_{auth}$ but cannot access the smart card. In a self-delegation scheme, if the adversary cannot distinguish the $K_{auth}$ from a random string of the same length with feasible computation, then the scheme is a secure self-delegation scheme.

We assume that the memory space of an adversary used for correcting delegated information is at most $poly(k)$, where $k$ is a security parameter. In this adversary model, the adversary is first given delegated information without $K_{auth}$, and either $K_{auth}$ or a random string depending on a coin flip $b$. Next, the adversary corrects the delegated information and outputs a guessed value of the coin flip $b'$. If the adversary's advantage $Adv_{delegation} = 2Pr[b = b'] - 1$ is negligible, the self-delegation scheme is a secure delegation scheme.

This definition implies that the adversary cannot obtain a correct pair of $K_{auth}$ and related information such as $R$ and $Ticket$ without the smart card. In the secret-key-based self-delegation, the adversary has to distinguish a correct tuple $(Uid, Info, R, K_{auth})$ without $Ks_U$. Where we assume $M(*)$ is a secure pseudorandom function, the advantage is negligible. Thus, the scheme is a secure self-delegation scheme. The security of the self-delegation reduces to pseudorandomness of function $M(*)$.

In the ticket-based self-delegation scheme, if the signature scheme is secure, the scheme is the same as the secret-key-based self-delegation scheme. Thus, the adversary cannot distinguish $K_{auth}$ with feasible computation.

Next, we discuss the *Forward Secrecy* 6),7) of self-delegation schemes. *Forward Secrecy* in the self-delegation defines the security of secondary key, when the primary key is compromised. This security notion is stronger than the Definition 1, because we assume an adversary who can obtain a primary information.

We define the forward secrecy of self-delegation as follows:

**Definition 2.** An adversary who knows primary information but cannot access the internal memory of a $PIC$ exists, and if he/she cannot distinguish $K_{auth}$ from a random number of the same length as $K_{auth}$ with feasible computation, then the self-delegation has *Forward Secrecy*.

An adversary's game is as follows: first, primary information and delegated information without $K_{auth}$ are given. In addition, either a random string or $K_{auth}$ are given depending on a coin flip $b \in \{0, 1\}$. An adversary corrects delegated information: then the adversary guess the coin flip and outputs the guessed value $b' \in \{0, 1\}$. If the advantage of the adversary, $Adv_{FS-delegation} = 2Pr[b = b'] - 1$, is negligible, then the delegation scheme has *Forward Secrecy*.

Generally, shared-key-based key exchange schemes cannot satisfy *Forward Secrecy*, because both entities have the same shared key and any secret communication is impossible if the condition of *Forward Secrecy* is assumed. In self-delegation, a secondary key has to be conveyed to the server secretly. Our secret-key-based self-delegation also does not have the *Forward Secrecy*. Thus, the scheme can be used only where the primary information is securely protected.

On the other hand, the ticket-based self-delegation has *Forward Secrecy* where an adversary cannot access the decryption function $D(*)$ with $SP$ as a oracle. In this situation, the adversary cannot obtain a correct pair of $< R, E(Kpu_{SP}, R) >$ with feasible computation, when the delegation uses a large-enough $R$ and a secure encryption scheme $E(*)$. Thus, the ticket-based self-delegation has forward secrecy because the adversary cannot test whether $K_{auth}$ is computed from $R$, that is, the adversary cannot checks the relation between $K_{auth}$ and $E(Kpu_{SP}, R)$ with feasible computation.

The two delegation schemes satisfy the other security requirements discussed in Section 3.3. The delegated keys depend on the primary key. The delegated key is computed using a one-way function. If the one-way function is secure, then the primary keys cannot be computed from delegated keys. The delegated information includes a period of validity. Thus, the delegated information expires at the end of the period.

## 4.2 Security Analysis of Authentication Protocols

In this section, we discuss the security of authentication protocols using delegated information. We prove the security of the authentication protocols based on the Bellare-Rogaway model[8)~10)]. The communication model proposed by Bellare and Rogaway is independent of the specification of the protocols, and the simulation of communication is constructed using oracles that define each entity and each session. An adversary controls all communications to interact with a set of oracles. The adversary can call oracles without a target oracle, and correct communication data of the protocol and all internal states of oracles. A secure mutual authentication protocol is defined as follows, where we describe a oracle $\Pi_X^i$ that simulates entity $X$ in session $i$.

**Definition 3.** If oracles $\Pi_{U_1}^i$ and $\Pi_{U_2}^j$ accept in the same session $(i = j)$, and the probability that one oracle (or both) accepts without an oracle that engaged in a *Matching Conversation*, then the protocol is a secure mutual authentication protocol.

Under the condition of *Matching Conversation*, the adversary only transfers communication between the oracles honestly without any interception and alteration. Thus, the adversary's strategy is geither uessing the correct authentication code, using an expired code, or using some other fresh authentication code that is received from a non-partnering entity. We define the probability of success without *Matching Conversation* is $Pr[Succ^{\text{No-Matching}^{E(k)}}]$. $Pr[Succ^{\text{No-Matching}^{E(k)}}]$ intuitively means the probability that an attack is successful without breaking a cryptographic function. If the probability is negligible, then the probability of breaking the authentication protocol is almost same as the probability of breaking a primitive cryptographic function. Bellare and Rogaway proved that the security of their authentication protocol is almost equal to the security of a pseudorandom function. We prove the security of our authentication protocols using the same method.

First, we prove the security of the secret-key-based authentication protocol.

**Theorem 1.** The secret-key-based authentication protocol is a secure mutual authentication protocol provided that $M(*)$ is assumed to be a pseudorandom function.

*Proof*. We first show the probability that the adversary $E$ is successful in the modified authentication protocol is negligibly small where $M(*)$ is replaced with a truly random function.

The probability that $\Pi_{SP}^s$ accepts without a matching conversation is calculated from the following three probabilities, where $\Pi_{SP}^s$ and $\Pi_{MT}^t$ is a partner.

- $Pr[Succ^{Reuse_{SP}}] = Pr[R'^s = R' | \{R'\} \in List]$ : *List* is the set of the communication log $\{*\}$ obtained from other oracles previously. This probability is that the adversary obtains a communication logs that includes the same as $\{R'\}$ in the first flow.

- $Pr[Succ^{Forge_{SP}}] = Pr[M^s(K_{auth}^s = \hat{M} | \hat{M} \overset{r}{\leftarrow} \{0,1\}^k]$: This probability is that the adversary can forge $M(K_{auth}, *)$ without $K_{auth}$

- $Pr[Succ^{InvK_{auth}}] = Pr[M^s(K_{auth}^s, *||R^s||*) = M(\hat{K}_{auth}, *||\hat{R}||*) | \hat{K}_{auth} \overset{r}{\leftarrow} \{0,1\}^k, \hat{R} \overset{r}{\leftarrow} \{0,1\}^k]$: This probability means the adversary can compute $M^s(*,*)$ using known/new pairs of $K_{auth}$ and $R$. In the initial condition of the protocol, $MT$ knows $K_{auth}$. However, $SP$ does not know $K_{auth}$. This attack is that the adversary persuades $SP$ to accept a $K_{auth}$ that is different from

**Table 1** Computational costs.

| | PIC | MT | SP |
|---|---|---|---|
| Delegation of a secret key | H | - | - |
| Delegation of a ticket | (E+S)+H | - | - |
| Secret-key-based authentication | - | 2H | 3H |
| Ticket-based authentication | - | 2H | 2V+D+3H |

a $K_{auth}$ of $MT$.

The probability that $\Pi^t_{MT}$ accepts without a matching conversation is calculated from the following two probabilities.

- $Pr[Succ^{Reuse_{MT}}] = Pr[R''^s = R'' | \{Uid, R, R'', Info, M(*)\} \in List]$ : This probability is that the adversary obtains a communication log that includes the same as $\{Uid, R, R'', Info, M(*)\}$ in the second flow.

- $Pr[Succ^{Forge_{MT}}] = Pr[M^s(K^s_{auth} = \hat{M} | \hat{M} \overset{r}{\leftarrow} \{0,1\}^k]$: This probability is that the adversary can forge $M(K_{auth}, *)$ without $K_{auth}$.

Let $T_E(k)$ denote the number of oracle calls with polynomial bound by the adversary $E$. $T_E(k)$ depends on the size of $List$ that the adversary can store. The total probability $Pr[Succ^{\text{No-Matching}^{E(k)}}]$ is calculated as follows, where the all random strings and $M(*) are in \{0,1\}^k$.

$$Pr[Succ^{\text{No-Matching}^{E(k)}}] \leq T_E(k) \cdot 2^{-k} + 2^{-k} + 2^{-k} + T_E(k) \cdot 2^{-k} + 2^{-k} = (2T_E(k) + 3) \cdot 2^{-k}$$

Thus, the probability that $M(*)$ is replaced with truly random functions is almost negligible.

Suppose that the probability the adversary is successful in the real authentication protocol which uses pseudorandom functions instead of truly random functions is not negligible. Then, we can construct a polynomial time test $T$ that distinguish truly random functions from pseudorandom functions. The construction of $T$ is the same as Bellare's proof. This result contradicts the pseudorandomness of $M(*)$. Thus, the secret-key-based authentication protocol is a secure mutual authentication protocol. □

Next, we discuss the security of the ticket-based authentication protocol.

**Theorem 2.** The ticket-based authentication protocol is a secure mutual authentication protocol provided that the $M(*)$ is assumed to be a pseudorandom function.

If we assume an adversary who cannot access the decryption function of the target oracle $\Pi^s_{SP}$, and the $PIC$ and the signature scheme are secure, then the case of the ticket-based authentication protocol is almost the same proof as the secret-key-based authentication protocol, because the adversary cannot obtain and replace $K_{auth}$ with feasible computation. Thus, we omit the detailed proof for the ticket-based protocol.

## 5. Evaluation

We evaluate proposed protocols in terms of data size of communication between the PIC and the MT, and the computational cost. In the evaluation, we use HmacSHA-1[11),12)] as a message authentication code algorithm, and RSA[13)] as a public-key encryption and signature algorithm.

We calculate the computational cost as shown in **Table 1**. H, E, D, S, and V indicate the cost of one hash calculation, one public key encryption, one decryption, one signing digitally, one verification of the sign, respectively. In secret-key-based self-delegation, computational costs of all entities are small. Ticket-based self-delegation is required for the heavy computation of public-key signing. If the PIC has computed a ticket previously, the computational cost (E+S) of the delegation can be eliminated. Thus, the two self-delegation schemes meet the requirements and conditions on services and the performance of the PIC. In the service authentication, computational costs for the MT are small.

**Table 2** shows a comparison of our scheme with existing self-delegation schemes. Goldreich, et al. proposed two self-delegation schemes. A generic scheme requires the computation of a non-interactive zero-knowledge proof

**Table 2** Comparison with existing self-delegation schemes.

| Schemes | Self-Delegation | Authentication (MT) |
|---|---|---|
| Goldreich [2] | NIZKP [17] | IZKP [18] |
| Goldreich (DL Scheme) [2] | two exponantations | ZKP [19] |
| Chaum [3] | at least E+S | at least E |
| Proposed (Secret-key-based) | H | 2H |
| Proposed (Ticket-based) | (E+S)+H | 2H |

(NIZKP) in self-delegation phase, and the computation of an interactive zero-knowledge proof (IZKP) is required for the mobile terminal. These computational costs seem to be much heavier than proposed schemes (a detailed calculation can be found in papers [17],[18]), and the scheme is not applicable to the PIC and the MT in terms of computational costs and data size. The transaction time of NIZKP, IZKP, and other zero knowledge proofs depends on a security parameter such as a number of rounds. Another scheme, named the Discrete-Logarithm (DL) scheme, is a more practical scheme based on discrete logarithms. In self-delegation, the PIC computes two exponentiations that require the computational cost to be the same as that of two signings. The cost is assumed to be about twice as the cost of our proposed scheme based on the ticket. However, the authentication scheme uses a zero-knowledge proof (ZKP) that is heavier than the proposed scheme, because the cost of our proposed authentication scheme is negligibly small. Chaum's scheme has similar computational costs to our proposed scheme, even though the authentication is slightly heavier. The transaction time for self-delegation and authentication is assumed about 2,000 msec in self-delegation, and 300 msec in authentication, respectively from the comparison with our evaluation results. However, the secondary information has to be protected against replay attacks in his scheme, because his scheme sends the secondary information to the SP. Thus, constructing of a secure channel is also required in the authentication phase. It may not be impossible to apply Goldreich's DL scheme and Chaum's scheme to the PIC and the mobile terminal; however, our proposed schemes have a lower computational cost.

We implemented the self-delegation and authentication protocols on a PDA and a Java card, both of which have ISO/IEC 14443 I/F [4]. A picture of the prototype system is shown in **Fig. 1**, and the structure of the self-delegation application is shown in **Fig. 2**. We also implemented cryptographic library and ISO 14443 communication functions for the application on Java runtime environments. The Java runtime environment of the MT is based on the MIDP 1.0 [5] environment for cellular phones.

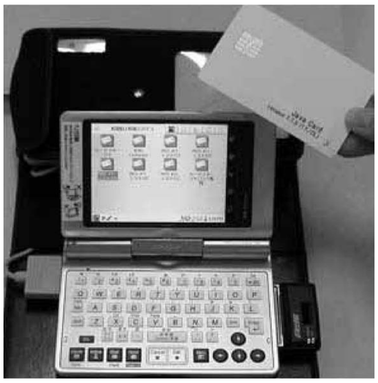

**Fig. 1** Picture of prototype system.

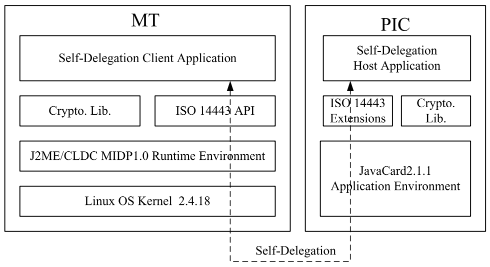
MT
Self-Delegation Client Application
Crypto. Lib.
ISO 14443 API
J2ME/CLDC MIDP1.0 Runtime Environment
Linux OS Kernel 2.4.18
PIC
Self-Delegation Host Application
ISO 14443 Extensions
Crypto. Lib.
JavaCard2.1.1 Application Environment
Self-Delegation

**Fig. 2** Structure of self-delegation application.

**Table 3** Evaluation of the Prototype System.

| Protocols | Tran. Time (msec) |
|---|---|
| Delegation of a secret key | 109 |
| Delegation of a ticket | 1,353 |
| Secret-key-based authentication | 44 |
| Ticket-based authentication | 211 |

The evaluation results are shown in **Table 3**. We use a PIC that has Java runtime environments and a 16-bits CPU, an MT that has a 400 MHz CPU and 64 Mbytes of memory, and a server of SP that has a 2.0 GHz CPU and 1.0 GBytes of memory. The transaction time for our self-delegation schemes is about 1 second. The authentication protocols are finished within 300 msec.

We conclude that the proposed protocols are applicable to mobile terminals and smart cards.

## 6. Application Examples

In this section, we describe examples of applications using self-delegation. One application is credit-card-based mobile commerce. A user performs the self-delegation from a couple of credit cards to their mobile terminal, and then the user carries the mobile terminal. The user buys items using delegated information for real or virtual commerce. The delegated information from the credit card can be applied for temporary use of a mobile terminal. When a user goes overseas and uses a rental cellular phone, they can easily pay the charge of the cellular phone using delegated information.

Another application is for personal identification. Assume that you have a public identification card; one problem with a public identification card is that the card may become lost. Delegating information for identification enables users to store the card in their home securely. Self-delegation is also applicable for temporary personal identification services. When a customer visits an office, temporary identification cards with short-term validity are often issued. Instead of issuing a card, temporal information for identification can be delegated to the customer's cellular phone using self-delegation.

Another application is personalization of cellular phones. Cellular phones are currently considered secure devices. Information for service use is stored directly on the cellular phone, and a special device is required to store the information. If a user changes to new cellular phone, they have to visits a shop that has a special system for personalization. Personalization of new terminals is problematic for management of users and terminals. For example, in 3GPP2[14], the Over-The-Air Service Provisioning scheme (OTA-SP) is currently being discussed. OTA-SP is an online personalization scheme using wireless networks. Information for authentication is downloaded from a server to a new mobile terminal. However, a secure channel between the server and the mobile terminal has to be constructed.

The other solution is using a removable SIM or UIM. SIM and UIM are tamper-proof devices and can be attached to and removed from the cellular phones. If the information is stored on the SIM or UIM, a user can transfer the information by removing and attaching the device. However, this scheme is also problematic. As mentioned in Section 2, if the cellular phone is stolen or lost, the information is also lost or open to exploitation.

The proposed scheme provides an offline personalization scheme, because all (primary) information is stored in the user's PIC. If a user purchases a new mobile terminal, the user only need to delegates their own authority to the mobile terminal on a local basis, using their own PIC. The personalization process using the self-delegation is also applicable to a PC and other devices such as a Set Top Box (STB), which has access to networks and several authenticated services.

Our self-delegation is applicable not only to the proposed authentication protocols but also to current authentication protocols such as SSL[15] by modifying the protocols slightly.

## 7. Conclusion

In this paper, we proposed a self-delegation which is an authentication mechanism for mobile environments, especially assuming that the user has a mobile terminal and a tamper-resistant module such as a smart card. We presented two self-delegation schemes for symmetric key and asymmetric key authentication. We analyzed the security of the proposed schemes based on the adversary model, which was defined for self-delegation. In addition, the security of authentication protocols using delegated information was proved using the Bellare-Rogaway model. Furthermore, we implemented the proposed schemes and discussed their feasibility. We showed that the protocols are secure and applicable to some example applications. We believe the self-delegation improves the services available for mobile terminals.

## References

1) Harada, H., Kuroda, M., Morikawa, H., Wakana, H. and Adachi, F.: The Overview of the New Generation Mobile Communication System and the Role of Software Defined Radio Technology, *IEICE Trans. Commun.*, col. E86-B, No.12, pp.3374–3384 (Dec. 2003).

2) Goldreich, O., Pftzmann, B. and Rivest, R.L.: Self-Delegation with Controlled Propagation - or- What If You Lose Your Laptop, *Proc. Crypto 98*, Springer LNCS, Vol.1462, pp.153–168 (1998).

3) Chaum, D.: Showing credentials without iden-

tification; Transferring signatures between unconditionally unlinkable pseudonyms, *Proc. Auscrypt'90*, Springer LNCS, Vol.453, pp.246–264 (1990).

4) ISO/IEC 14443, Identification cards–Contactless integrated circuit(s) cards–Proximity cards.
5) Sun Microsystems.: Mobile Information Device Profile (MIDP) 1.0.
6) Diffie, W., van Oorschot P. and Wiener, M.: Authentication and authenticated key exchanges, Designs, *Codes and Cryptography*, Vol.2, pp.107–125 (1992).
7) Gunther, C.G.: An identity-based key exchange protocol, *Proc. EUROCRYPT'89*, LNCS, Vol.434, pp.29–37 (1990).
8) Bellare, M., Rogaway, P.: Entity authentication and key distribution, *Proc. CRYPTO'93*, LNCS, Vol.773, pp.232–249 (1994).
9) Bellare, M. and Rogaway, P.: Provably secure session key distribution: the three party case, *Proc. 27th Annual Symposium on the Theory of Computing*, ACM, pp.57–66 (1995).
10) Bellare, M., Pointcheval, D. and Rogaway, P.: Authenticated Key Exchange Secure Against Dictionary Attacks, *Proc. Eurocrypt'00*, LNCS, Vol.1807, pp.139–155 (2000).
11) Krawczyk, H., Bellare, M. and Canetti, R.: "HMAC" Keyed-Hashing for Message Authentication, RFC 2104 (1997).
12) NIST.: Secure Hash Standard, FIPS PUB 180-1 (1995).
13) Rivest, R.L., Shamir, A. and Adleman, L.M.: A Method for Obtaining Digital Signatures and Public-key Cryptosystems, *Comm. ACM*, Vol.21, pp.120–126 (1978).
14) The 3rd Generation Partnership Project 2 (3GPP2). http://www.3gpp2.org/
15) Freier, A., Karlton, P. and Kocher, P.: The SSL Protocol Version 3.0. http://home.netscape.com/eng/ssl3/draft302.txt
16) Fox, A. and Gribble, S.: Security On the Move: Indirect Authentication Using Kerberos, *Mobile Computing and Networking*, pp.155–164 (1996).
17) Blum, M., Feldman, P. and Micali, S.: Non-Interactive Zero-Knowledge and its Applications, *Proc. 20th STOC*, pp.103–112 (1998).
18) Goldwasser, S., Micali, S. and Rackoff, C.: The Knowledge Complexity of Interactive Proof Systems, *SIAM Journal on Computing*, Vol.18, No.1, pp.186–208 (1989).
19) Fiat, A. and Shamir, A.: How to Prove Yourself: Practical Solutions to Identification and Signature Problems, *Proc. Crypto'86*, LNCS, Vol.263, pp.186–194 (1987).
20) Yoshida, M., et al.: A Secure Service Architecture for Beyond 3G Wireless Network, *WPMC 2003* (Oct. 2003).

(Received November 1, 2004)
(Accepted June 9, 2005)
(Online version of this article can be found in the IPSJ Digital Courier, Vol.1, pp.282–293.)

**Shinsaku Kiyomoto** received his B.E. in engineering sciences, and M.E. in materials science, from Tsukuba University, Japan, in 1998 and 2000 respectively. He joined KDD (now KDDI) and has been engaged in the research on stream cipher, cryptographic protocol, and mobile security. He is currently a research engineer of the Security Lab. in KDDI R & D Laboratories Inc. He received the Young Engineer Award from IEICE in 2004. He is a member of JPS and IEICE.

**Toshiaki Tanaka** received the B.E. and M.E. degrees in communication engineering from Osaka University, Japan, in 1984 and 1986 respectively. He joined KDD (now KDDI) and has been engaged in the research on cryptographic protocol, mobile security, digital rights management, and intrusion detection techniques. He is currently a senior manager of the Security Lab. in KDDI R & D Laboratories Inc. He is a member of IEICE.

**Mariko Yoshida** received the B.S. degree from the Tokyo Institute of Technology, Japan, in 1994. She joined Mitsubishi Electric Corporation, Japan in 1994. Since then, she was engaged in business computer systems, mobile applications. Her current research interests includes wireless security and wireless applications.

**Masahiro Kuroda** received the M.E. degree in systems science from the Tokyo Institute of Technology, Japan, in 1980, the M.S. degree in computer science from University of California, Santa Barbara, CA, in 1989, and received the Ph.D. degree in computer science from Shizuoka University, Japan, in 2000. He joined Mitsubishi Electric Corporation, Kamakura, Japan in 1980. Since then, he was engaged in OS/network developments, mobile network computing R & D, and cellular Java standardizations. He is currently working as a group leader at National Institute of Information and Communications Technology, Yokosuka, Japan. His current research interests includes wireless network, wireless security, mobile systems, and next generation wireless systems architecture. He is a member of the IPSJ and IEEE Computer Society.