User Authentication with the Help of Simple Memory Devices

Keiji Amo Yuichi Kaji Tadao Kasami

{keiji-a,kaji,kasami}@is.aist-nara.ac.jp

Graduate School of Information Science
Nara Institute of Science and Technology

Takayama 8916-5, Ikoma, Nara 630-01, Japan

abstract We propose a simple and secure user authentication method with the help of simple memory devices such as magnetic cards. In this method, user's password is divided into two pieces; one is kept in the user's mind and the other is stored in the memory device. The original password is reconstructed if and only if both of them are presented. The contents of the memory device are encrypted so that the robbery of the card does not effect to the security of the method. Furthermore, the contents of the device are changed in every session, which makes the method robust against replay attacks and minimizes the damage of the user in case of an emergency. Using this method, we can easily overcome the problem of impersonation of an intruder caused by network attacks, a fault of users, illegal copy of the card and so on. We give the definition of the method and discuss its security.

1 Introduction

Nowadays, various personal authentication methods have been proposed. When one discusses the security of a personal authentication method, there are at least two points that should be taken into consideration. The first point is how to distinguish a user from other users, and the second point is how the information for authentication is exchanged through (possibly insecure) communication channels. With respect to the first point, we can classify authentication methods into some classes. The simplest class consists of methods that use only secret information such as passwords and PINs (personal identification numbers) to distinguish a person from others[2]. The merit of this class is that the system becomes simple and economical. On the other hand, the security is not very strong since if an evil party happens to know the secret information, then there is no way to prevent impersonation. The second class consists of methods that use one's belongings such as ID cards[8]. This method, used together with passwords, is considered to be practically secure, and actually is adopted by many banks as the authentication method in their cash-card system. The last class consists of methods which use biometrics of users, such as fingerprints and voice. However, this class is still in developing, and the system will be very expensive.

By taking costs, simplicity and security into consideration, the authors conjecture that authentication methods that use secret information and belongings (i.e. passwords and cards) will continue to play a significant role. Therefore, it is worthwhile to improve the security of this (passwords plus cards) method. Conceptually, in such a method, the information necessary for authentication is divided into two parts. One is remembered by a user, and another is stored in the user's belongings such as a magnetic card. The former part is called a *user share* and the latter is called *memory share* in this paper. The original information should be reconstructed if and only if both of the user share and the memory share are presented. For the sake of security, we must pay attention to how the secret information is divided. If the division of the secret is carried out so that user shares play too significant role in authentication, then the security will not be strong since user shares tend to be short and easily inferable, and it is not very difficult for an intruder to examine all candidates of user shares. Conversely, if the division is such that memory shares play too much role in authentication, then system will be fragile for the robbery and illegal copy of cards. To the author's knowledge, there has not been sufficient discussion on how such division should be carried out.

In this paper, an authentication method that uses secret information and user's belongings is proposed. In the proposed method, the user share is a short string that is easy to remember, and the memory share is a long and random string that no one can guess. The shares are chosen independently so that knowledge on one share does not give any information of the other share. Thus, even if an intruder obtains either one of them by peeping or by robbery, he/she cannot impersonate the user by any means. Furthermore, in the proposed method, the memory share is changed every time the card is used[3][4]. This

feature minimizes the damage in an emergency case such that an intruder obtains both of the user share and the memory share.

Consider the second point to be taken into consideration, that is, how the information for authentication is exchanged through communication channels. According to the development of open networks such as the Internet, the authentication via an insecure communication channel is an emergent matter. Thus authentication methods must be robust against network attacks such as wiretapping, replay attacks, modification of messages, server pretenders, dictionary attacks and exhaustive attacks. The proposed method is robust against all of the above attacks.

2 Imaginable Environment and Attacks

In the following discussion, we consider the network model illustrated in Fig 1. We call the machine which a user uses *terminal* and the authentication machine which has a password file *server*. It may happen that an intruder wiretaps the communication, forges messages, modifies messages on the communication channels, pretends to be the server and steals (copies) a password file from the server. It is assumed that each user remembers one's user share and posses an ID card in which one's memory share is stored. Generally speaking, the user share may be a short, simple and possibly easy to guess word, while the memory share can be a relatively longer and random string.

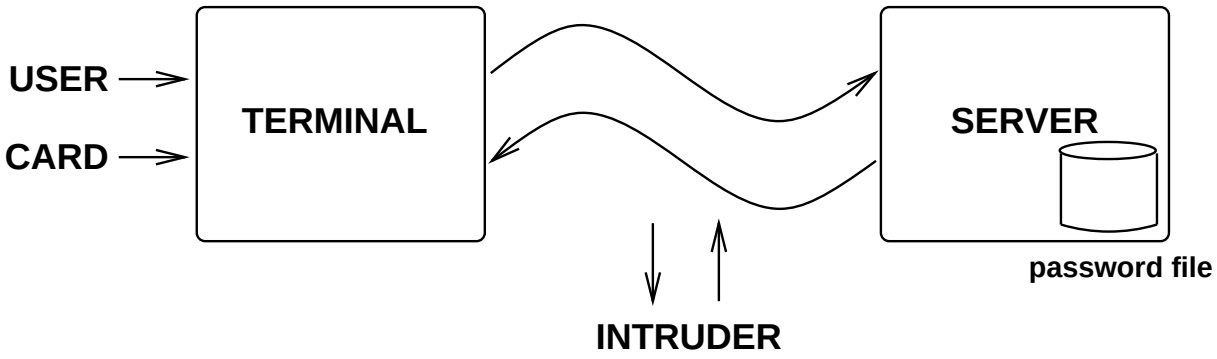


Figure 1: A model of the network

The purpose of the protocol is that, if the protocol is carried out between a legitimate user with a correct ID card and the server, then he/she is able to authenticate himself/herself to the server. Furthermore, the following attacks by an intruder should not be succeeded.

- wiretapping

An intruder wiretaps the communication between the terminal and the server, and tries to obtain the information that can be used to impersonate other users (e.g. replay attack). The information obtained by wiretapping might be used in all of the attacks described below. Wiretapping and replay attacks can break the naive telnet protocol of UNIX operating system.

- message modification

  An intruder modifies, exchanges, or forges messages on the communication channel to impersonate other users or to obtain the information that can be used to impersonate other users. If the message modification attack is forced on S/Key system[2] at the key re-initialization step, then the pass-phase chosen by an intruder can be registered in the server as the legitimate user's pass-phase.

- pretending to the server

  An intruder plays the role of the server, and tries to steal the information for impersonation. This attack is effective to many challenge-response authentication protocol including S/Key.

- dictionary attack

  An intruder guesses the information used for authentication and checks whether the guesses are correct or not. According to the way to check the correctness of guesses, the dictionary attack can be classified into two attacks[5]. In an **"on-line" dictionary attack**, the correctness of the guesses are checked by interacting with the server through the terminal. Therefore, the on-line dictionary attack will not be great threat since we can detect the attack by observing repeated failure of authentication. On the other hand, in an **"off-line" dictionary attack**, the intruder himself/herself encrypts the guesses and compares the ciphertexts of the guesses with entries in a password file which he/she has stolen from the server (this is possible in some old UNIX system), or with the ciphertext which is obtained by wiretapping. Thus, the off-line dictionary attack is not detected by the server and therefore can be greater threat than the on-line attacks.

- exhaustive attack

  Similar to the dictionary attack, but an intruder tries all possible passwords (or secret information) one by one. In recent years, computers are getting faster and faster, and therefore, the exhaustive attack will be great threat if the domain of the secret is not very large.

- peeping the password

An intruder who happens to obtain other user's share tries to impersonate the user. Unfortunately, the possibility for an intruder to obtain other user's share is not small. Some users who do not pay much attention on the security sometimes write and display their password near their computer. Most authentication systems without additional physical devices are helpless if the password (user share) is known to the intruder.

- duplicating the card

  An intruder obtains the contents (memory share) of an ID card, and tries to impersonate the user. Since the contents of magnetic cards used in many systems are easily readable, it is not difficult for an intruder to duplicate the contents of the card. The situation will be better if an IC-card[8] is used instead of magnetic cards, though, the situation is essentially the same unless ideal tamper-resistant modules are used. If the card plays too significant role in an authentication, then robbery or illegal duplication of the card can be great threat.

3 Formal Definition

In the following, we describe the formal definition of the authentication method. The method is divided into two phases, user registration phase and user authentication phase.

3.1 User Registration Phase

A user who wishes to use this system must be registered first. In this registration step, we assume that there is a (possibly off-line) secure channel between the user and the server. Remark that this is not impractical assumption. For example, remind the process to open a bank account. We go to a branch of the bank, decide the password number for a cash-card and write it down on an application form. The form is sent to the head office of the bank via a "secure" channel, and a cash-card is sent back to the user by a "secure" registered letter. We assume the existence of a secure channel of this kind. It is also assumed that the server and the terminals agree on a symmetric encryption function to be used in the system, and the server has a secret key K which no one else knows.

Step 1 A user, say Alice, chooses u as her user share, and informs the server that she has chosen u via the "secure" channel.

Step 2 The server receives u from Alice, randomly chooses m as her memory share and computes $e(K, u||m)$ where $||$ stands for string concatenation and $e(x, y)$ denotes the ciphertext of y encrypted by x as a key. The server records a pair $\langle$"Alice", $e(K, u||m)\rangle$

in the password file, computes $e(u, m)$, writes down "Alice" and $e(u, m)$ on a memory device (e.g. magnetic card), and sends it to Alice via the "secure" channel.

3.2 User Authentication Phase

To check whether a user is valid or not, the following protocol is executed. Roughly speaking, the protocol consists of two phases. In the first phase (Step 1–4), a virtual secure channel is established by using a variant of Diffie-Hellman key exchange protocol[1]. Message exchanges for actual authentication are carried out in the second phase (Step 4–6). It is assumed that a large prime number q and a primitive element α of $GF(q)$ are selected and published where $GF(q)$ is a finite field with q elements.

Step 1 A user, say Alice, inserts her memory device to the terminal, and inputs her user share u'.

Step 2 The terminal reads the user's name "Alice" and (the ciphertext of) memory share $e(u', m')$ from the inserted memory device. Then the terminal receives her user share u' from its keyboard, uses it to decrypt $e(u', m')$ and obtains m'. Furthermore, the terminal randomly chooses an integer X_T such that $1 \leq X_T \leq q - 1$. The terminal then computes $Y_T = \alpha^{X_T} mod\ q$, and sends "Alice"$||e(m, Y_T)$ to the server.

Step 3 The server receives the user's name "Alice" and retrieves $\langle$"Alice", $e(K, u||m)\rangle$ from its password file. The server decrypts $e(K, u||m)$ by its secret key K, and obtains u and m. The server decrypts the received message by using m and obtains Y_T. Then the server randomly chooses X_S such that $1 \leq X_S \leq q-1$, computes $Y_S = \alpha^{X_S} mod\ q$, and sends back $e(m, Y_S)$ to the terminal. Furthermore, the server computes $SK = Y_T^{X_S} mod\ q\ (= \alpha^{X_T X_S} mod\ q\ = \alpha^{X_T X_S} mod\ q)$ which will be used as a session key.

Step 4 The terminal decrypts the received message $e(m, Y_S)$ by using m' and obtains Y_S. Then the terminal computes $SK = Y_S^{X_T} mod\ q (= \alpha^{X_S X_T} mod\ q\ = \alpha^{X_T X_S} mod\ q)$ as a session key. Since both of the server and the terminal (user) know correct m, they have had an identical session key SK at this time. The terminal computes $e(SK, u'||m')$ and sends it to the server.

Step 5 The server decrypts the received message $e(SK, u'||m')$ by using SK and retrieves u' and m'. If $u' = u$ and $m' = m$ where u and m are retrieved from the password file in step 3, then the user is considered to be Alice. In this case, the server randomly chooses m'' as the next memory share, computes $e(K, u||m'')$ and replaces $e(K, u||m)$ in the password file with $e(K, u||m'')$. Furthermore, the server computes $e(SK$,"Alice is authenticated"$||e(u, m''))$ and sends it back to the terminal. If $u' \neq u$ or $m' \neq m$, then the user cannot be Alice, thus the server compute $e(SK$, "Alice is not authenticated") and sends it back to the terminal.

Step 6 If the terminal receives the message "Alice is authenticated", then it replaces $e(u, m')$ in the memory device with $e(u, m'')$, and the authentication is successfully complete. The consequent communications between the server and the terminal will be encrypted by using SK as a session key.

4 Security of the Method

In this section, we describe that the method is robust against attacks described in section 2.

The primal purpose of an intruder, Mallet, is to find both of Alice's user share u and the ciphertext $e(u, m)$ of her memory share. In the following, we sketch that it is impossible for Mallet to obtain both of u and $e(u, m)$.

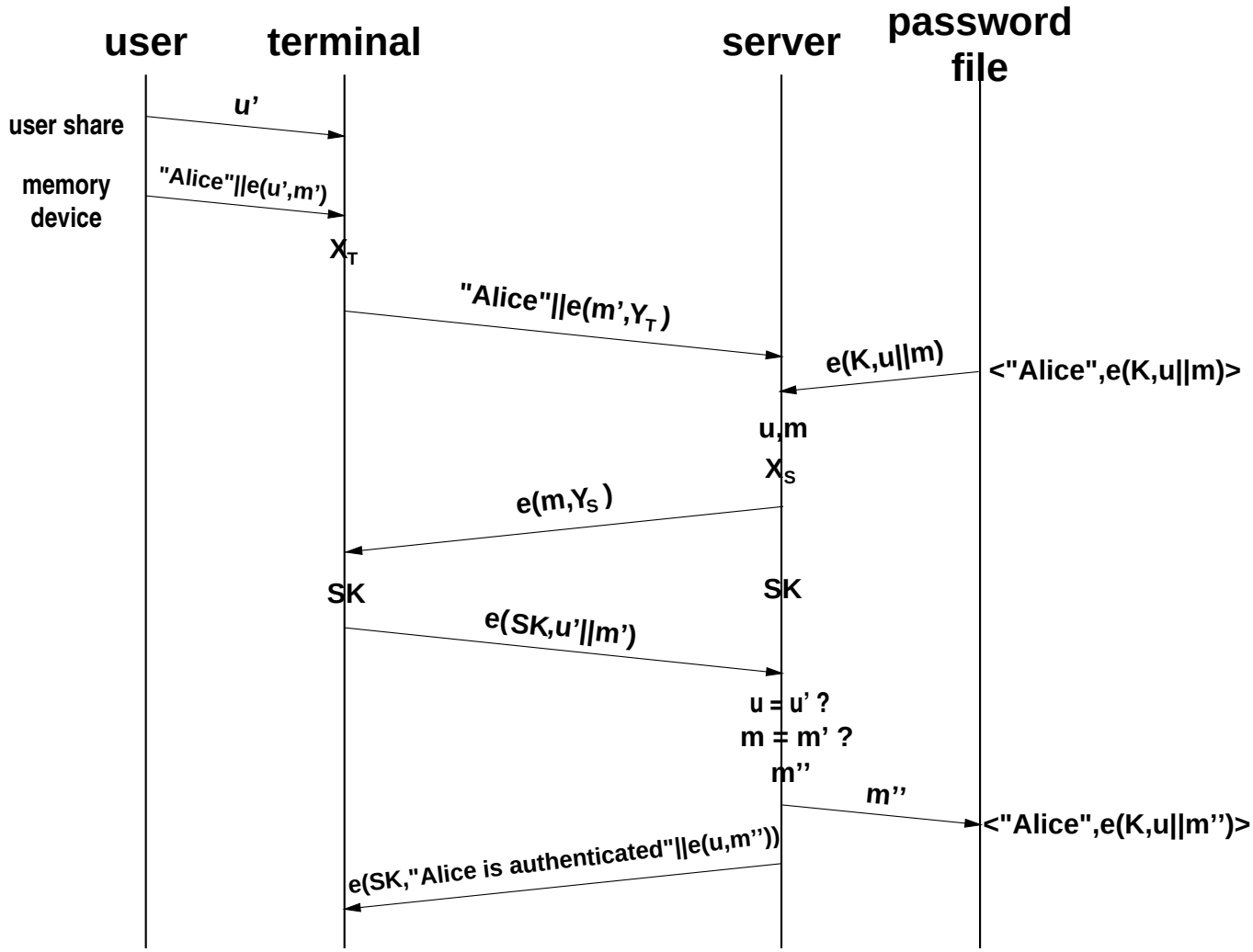
Lemma 1 *Mallet cannot obtain both of u and m by wiretapping, off-line dictionary attacks and exhaustive attacks.*

Proof: To obtain u and m by wiretapping Mallet has to know the session key SK, which is doubly protected. The first protection is that the first and the second messages exchanged between the terminal and the server are encrypted by m which Mallet does not know. Hence he cannot obtain Y_T and Y_S which are necessary to compute SK. Even if Mallet happens to know the protected Y_T and Y_S, it is very hard to compute SK. Computing SK is equivalent to breaking the Diffie-Hellman key exchange protocol[**?**], and its difficulty is based on that of solving discrete logarithms. This is the second protection, and therefore Mallet cannot obtain u and m by wiretapping.

Next, we consider exhaustive attacks and off-line dictionary attacks in addition to wiretapping. The information which Mallet can refer for these attacks is $e(K, u||m)$, $e(m, Y_T)$, $e(m, Y_S)$, $e(SK, u||m)$, $e(SK$, "Alice is authenticationed"$||e(u, m''))$ and other messages encrypted by SK. The information $e(K, u||m)$ is the record of the password file and the others are obtained by wiretapping. However, since Mallet does not know K, m and SK, he cannot encrypt his guesses, and cannot judge which guess is correct. Though he may try to find SK by exhaustive attacks against $e(SK$,"Alice is authenticationed"$||e(u, m''))$, the probability to succeed is very small since SK is long and random string. □

Lemma 2 *Even if Mallet happens to know u, he cannot obtain m by wiretapping, off-line dictionary attacks and exhaustive attacks.*

Proof: Even if Mallet knows u, the argument of Lemma1 holds. A little attention should be paid for exhaustive attacks against $e(K, u||m)$ and $e(SK, u||m)$. If Mallet knows the decryption procedure, then, for every candidate of K or SK, he may decrypt these messages



e : encryption by a symmetric key cryptography
K : the public key of the server
SK : a session key

Figure 2: A protocol of user authentication

by the candidate and check whether the deciphered message has u as its prefix. Thus, theoretically, the exhaustive attack is possible. However, the probability to succeed is very small since K and SK can be a very long string (since a user do not have to remember them), K is selected by the server carefully and SK is selected randomly in every session. □

Lemma 3 *Even if Mallet happens to know the ciphertext $e(u, m)$ of Alice's memory share, he cannot obtain u and m by wiretapping, off-line dictionary attacks and exhaustive attacks.*

Proof: Even if Mallet knows $e(u, m)$ in Alice's memory device, the argument of Lemma 1 holds. However, in this case, we must pay attention to the exhaustive attacks against u. Since u is the information that human remembers, the domain of u is small. Mallet may be able to enumerate all candidates $u_1, \cdots, u_n$ of u. For each u_i, he can decide m_i such that $e(u, m) = e(u_i, m_i)$, Y_{Ti} such that $e(m, Y_T) = e(m_i, Y_{Ti})$ and Y_{Si} such that $e(m, Y_S) = e(m_i, Y_{Si})$. However, since m_i, Y_{Ti} and Y_{Si} are all random, he cannot decide which u_i is the correct guess of u. □

If the contents of the device are not encrypted, then robbery of the device immediately gives Mallet m. This situation will be critical in some environments, for example discussed in lemma 4.

Lemma 4 *Even if Mallet pretends to the server and exchanges messages with the terminal, he cannot obtain both of u and m.*

Proof: Mallet intercepts $e(m, Y_T)$ which is sent by the terminal, and sends back a message Z_S instead of the server. However, since Mallet does not know m, there is no guarantee that Z_S is a valid (i.e. decryptable by m) ciphertext. If Z_S is not a valid ciphertext, then the protocol ends abnormally and Mallet cannot obtain secret information. Even if Z_S is a valid ciphertext, Mallet cannot know X_S $(1 \leq X_S \leq q-1)$ such that $Z_S = e(m, \alpha^{X_S})$. On the other hand, the terminal computes $SK (= {Z_S}^{X_T} = \alpha^{X_S X_T})$ and sends back $e(SK, u||m)$. However, Mallet cannot compute SK, hence he cannot obtain u and m. □

In the proposed protocol, it is very important that the first and the second messages exchanged between the terminal and the server are encrypted by m. If these messages are not encrypted by m, that is, if we use the original Diffie-Hellman key exchange protocol[1], then Mallet can obtain u and m by an intruder-in-the-middle-attack[7] described as follows. First, he intercepts Y_T sent by the terminal. Then he chooses X_M $(1 \leq X_M \leq q-1)$, computes $Y_M = \alpha^{X_M} mod\ q$ and sends Y_M to the terminal. The terminal and Mallet will share the same session key $\alpha^{X_M X_T}$ and Mallet will decrypt $e(\alpha^{X_M X_T}, u||m)$.

The above lemmas imply that Mallet cannot obtain u and m. Even if Mallet cannot know u and m, he may try the replay attacks by using information which he has obtained

by wiretapping. However, as the following lemma shows, the probability that the replay attack succeeds is very small. Moreover, it is very hard for Mallet to obtain a session key SK at the same time.

Lemma 5 *The probability that the replay attack succeeds is very small. Moreover, even if the replay attacks succeed, Mallet cannot obtain SK.*

Proof: Suppose that the authentication protocol have been executed n times before. The information that Mallet has obtained by wiretapping is $e(m, Y_{Ti})$, $e(m, Y_{Si})$, $e(SK_i, u||m_i)$, $e(SK_i$, "Alice is authenticationed"$||e(u, m_{i+1}))$ $(1 \leq i \leq n)$ where Y_{Ti} and Y_{Si} are the values which are selected by the terminal and the server, respectively, in the i-th session, SK_i is the i-th session key and m_i is the memory share which is used in the i-th authentication (therefore, the current valid memory share is m_{n+1}). The replay attack succeeds if and only if there is i $(1 \leq i \leq n)$ such that $m_i = m_{n+1}$ and when Mallet sends $e(m_i, Y_{Ti})$ to the server, the server sends back $e(m_{n+1}, Y_S)$. This probability is very small, and even if such a situation happens, since Mallet does not know SK_i, he cannot understand messages sent by the server. □

We have discussed the security of the proposed protocol. In addition to the security robustness, the method has the following favorable features.

- Sensitive to accidents and cheatings.

  As we have seen in lemmas 2 and 3, even if either one of Alice's user share or her memory device is stolen or copied by Mallet, the system is left secure. If Mallet obtains both of Alice's user share and her memory device, then he can impersonate Alice. This is essentially unavoidable, but quite undesirable. In many authentication methods, Alice will not find this accident until she notices an unconvincing bill, which is often too late. However, by using the proposed method, Alice can detect Mallet's cheating easier and can minimize the damage. Assume that there are two cards of which contents are the same, one is a valid card of Alice and the other is an illegal card duplicated by Mallet. If Alice uses her proper card before Mallet uses his illegal card, then the memory share m is changed. According to this change, the illegal card of Mallet is invalidated and Mallet cannot use it. If Mallet uses his card before Alice uses her card, then her valid card is invalidated. Hence, she can find this emergency situation and can close her account promptly. Furthermore she can reason anyone who made such an illegal duplication, since such an duplication must be made after she has successfully used her card before.

- Secret information is very small and concentrated.

  In our method, the secret information which must be guarded by the system is only the secret key K of the server. Terminals do not have to keep any secret information.

This can be a great advantage in an open network in which there are many terminals with inferior security level. It is also favorable in a situation such that a terminal is located in a dangerous place and may receive physical attacks. Since it is very expensive to realize a mechanism to guard information from such physical attacks, the system will be expensive if terminals have to keep secret information.

- User-interface is simple and easy.

  A user only have to remember a short and simple user share u, and carry the simple memory device (e.g. magnetic card). Since this style is the same as that of a typical cash-card system of a bank, the method is very familiar to many people. Moreover, the method does not rely too much on users. For example, one-time passwords such as S/KEY system functions very well as far as users pay enough attention to the usage of the system. However, it is not practical to require all users to have such attentions. On the other hand, users who are using cash-cards can easily accept our method without any consciousness. Furthermore, even if he/she does not pay attention on the security of the system, stronger security is guaranteed in our method.

- Only a symmetric key cryptography is used.

  If we use a public key cryptography in the protocol, then the protocol can be simpler than the proposed one[4]. However, in that case, the problem of how to deliver correct public keys issues. Though some solutions to this problem has been investigated[6], the delivery of public keys is a difficult problem, and a large and complex mechanism will be necessary to realize secure key derivation in general. Therefore, we used a symmetric key cryptography, which makes the system more practical.

5 Summary

The user authentication method with the help of simple memory devices is proposed, and its security, features and merits are discussed. The proposed method is robust against wiretapping, replay attacks, modification of messages, server pretenders, dictionary attacks and exhaustive attacks. Moreover, it is robust against faults of a user such as theft of his/her memory device, leak of his/her user share, a wrong choice of his/her user share and so on. Even if both of his/her user share and his/her memory device are stolen, the damage can be minimized since the contents of the device are changed in every session. In the proposed method, users do not have to pay extraordinary attention to the security of the method. Furthermore, the method has a simple and easy user-interface, which will be widely accepted by ordinary users. Therefore, the method is very suitable for the the bank system with cash-cards. By using the proposed method, the system will be robust against user's faults. Other promising application is a login scheme of a computer system (e.g.

UNIX operating system). If the proposed method is employed by using a floppy disk as a simple memory device, then we can economically realize a system which is robust against various attacks. The last example of applications is an EC (Electronic Commerce) through an open network such as the Internet. Since the proposed method is robust against network attacks and user's faults, it is suitable for this application.

References

[1] Diffie, W. and Hellman, M. E., "New directions in cryptography", IEEE Trans. Inform. Theory, **IT-22**, 6, pp.644–654 (1976).

[2] Haller, N., "The S/Key(TM) One-Time Password System", Proceedings of the Internet Society Symposium on Network and Distributed System Security, pp.151–158 (1994).

[3] Kaji, Y. and Kasami, T., "Declare-Next Authentication Method—Secure Use of Insecure Magnetic Cards", SCIS96-5D (1996).

[4] Keiji, A., Kaji, Y. and Kasami, T., " Authentication Method with the Assistance of Simple Memory Devices", SCIS97-19B (1997).

[5] Menezes, A. J., van Oorschot, P. C. and Vanstone, S. A., "*Applied Cryptography*", CRC Press (1996).

[6] Needham, R. and Schroeder, M., "Using Encryption for Authentication in Large Networks of Computers", Communications of the ACM, **21**, 12, pp.993–999 (1978).

[7] Stinson, D. R., "*Cryptgraphy*", CRC Press (1995).

[8] Zoreda, J. L. and Otón, J. M., "*Smart Cards*", Artech House (1994).