社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2004－DD－45　(3)
2004／7／30

自己完結型認証方式 SAIFU

辻　　貴介†　　清水　明宏††

† †† 高知工科大学　〒 782-8502 高知県香美郡土佐山田町宮ノ口 185
E-mail: †076035k@gs.kochi-tech.ac.jp, ††shimizu.akihiro@kochi-tech.ac.jp

あらまし　インターネットやモバイル通信が普及し，個人情報を扱うサービスが増加している．そのようなシステムでは，アクセス制御が不可欠であり，ユーザの権限を保証するためにユーザ認証の技術が必要とされている．最近では，IC カードや携帯電話等を用いたユビキタスコンピューティング技術が発展し，そのような技術を用いた認証方式においては，機能性が重要である．そこで，我々は認証者に代わりエージェントが被認証者の資格を検証する新しい方式を提案する．

キーワード　認証，ワンタイムパスワード，ユビキタス，IC カード，携帯電話

Secure Agreement Identification for Flexible Users (SAIFU)

Takasuke TSUJI† and Akihiro SHIMIZU††

† ††Kochi University of Technology,
185 Miyanokuchi, Tosayamada-cho, Kami-gun, Kochi-ken, 782-8502, Japan
E-mail: †076035k@gs.kochi-tech.ac.jp, ††shimizu.akihiro@kochi-tech.ac.jp

Abstract　Internet communication applications for managing personal information have been developing. Access control systems are indispensable to those applications and need password authentications to protect users' right from attacker. One-time password authentication schemes are proposed in which the user changes the verifier in every session and is authenticated by the server though the Internet. Ubiquitous computing such as smart cards and mobile phones has been developing and functional authentication systems are desirable for ubiquitous networks. Here we propose a new authentication scheme in which agents verify the user instead of the server.

Key words　authentication, one-time password, ubiquitous, smart card, mobile phone

1. Introduction

The Internet and mobile communications have been developing, and related applications for managing money or personal information are increasing in number. However, there is a risk that such private data can be wiretapped. To counter this, those systems require an access control, in which user authentication is indispensable. This can be accomplished by the use of a one-way function with which it is easy to compute $f(x)$ from x and difficult to compute x such that $y = f(x)$. Usually hash functions such as MD5 [1] or SHA-1 [2] are employed.

Lamport's method [3] is a one-time password authentication method using a one-way function, and the S/Key authentication system [4], which is based on Lamport's scheme, have been proposed. The S/Key system is vulnerable to several attacks [7], [11] and has two practical difficulties: high hash overhead and the requirement of resetting the verifier. For those problems, the CINON (chained one-way data verification method) [5] and the PERM (Privacy Enhanced information Reading and writing Management method) [8] use a random number. However, those methods suffer from vulnerability to one kind of 'Man in the Middle' attack, and the SAS (Simple And Secure password authentication protocol) [9] has solved this weakness. C.L. Lin *et al.* have detected replay and denial of service attacks on the SAS method and have argued for the OSPA (Optimal Strong-

Password Authentication) [10] method. However, the authors have performed successful impersonation attacks on the OSPA method [12].

In the SAS, the user uses a one-way function five times. However, this function has high overhead, because a one-way function applies hash functions or common-key cryptosystems. It is desirable to reduce the number of times this function is used in the SAS because authentication methods are now being applied to mobile communications. Accordingly, the authors have proposed the SAS-2 (SAS version 2) [13], which reduces such overhead.

In the ubiquitous computing, authentication systems have to be more flexible because users want to use an authentication system for many agents, which mediate between the users and the server. This paper discusses problems of existing schemes when an authentication scheme is applied to pliable systems, in which many agents authenticate users in place of the server. Then, we propose a new authentication scheme.

2. Objective of Study

Many simple authentication schemes are proposed and those schemes are valuable for encryption communication on the Internet. However, those schemes are useless for ubiquitous identification systems, which demand real-time processing. The SAS-2 [13] scheme can be applied to electronic-lock systems, in which the user can open only one lock or few lock, which is connected with the authentication server in local area. In this scheme, the user cannot open the lock in other area.

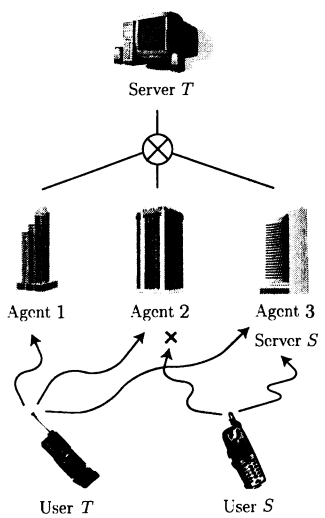

figure 1 Electronic-lock systems.

In the figure 1, User S can open the door, which is connected with Server S in local area. However, User S cannot open the door in other office. User S may open the doors in some offices, which are connected with the Server S by private networks. Although User S can open some doors, the door will not open in no time because the authentication data is sent Server S which is far away. Accordingly, functional authentication protocol is desirable.

In the figure 1, User T can open the doors in Agent 1, Agent 2, and Agent 3, which are connected with the Server T through the Internet. Then those agents identifies User T instead of Server T, and Server T synchronizes users' verifier. Hence, User T can open the door fast. This study aims to propose a new authentication scheme in which agents verify the user instead of the server.

3. Proposal Scheme

The secure authentication scheme SAIFU(Secure Agreement Identification for Flexible Users) is proposed. In this section, this scheme is illustrated.

3.1 Definitions and Notations

The following definitions and notations are used throughout this paper.

- U refers the computer user who employs the protocol for authentication.
- S refers to the server that authenticates users.
- A refers an agent who authenticates users in place of the server.
- ID_X is X's identity or login name.
- h is a conventional hash function. For example, $h(x)$ means x is hashed once.
- m is masking function (e.g. $m(x, y) = x \oplus y$).
- i is an integer indicating the number of authentication sessions.
- N_{Xi} represents X's random number corresponding to the ith authentication.
- $\oplus$ represents a bitwise exclusive OR (XOR) operation.
- $||$ denotes concatenation.
- G_{Xi} denotes $h(N_i)$ calculated by X, where $h(N_i)$ may be calculated from a combination of a random number and other data such as user's password.
- $M_i X_Y$ represents a masking data corresponding to the ith authentication, and X retrieves Y's masked information from this masking data.
- $H_i X_Y$ represents an authentication code corresponding to the ith authentication, and X authenticates Y using this authentication code.

- The expression '$s \Rightarrow t$: u' represents the fact that s sends u to t through a private, authenticated channel. For example, s delivers to t directly or encrypts and sends u with certification.
- The expression '$s \rightarrow t$: u' represents the fact that s sends u to t through a common channel such as the Internet.

3.2 The SAIFU Protocol

The SAIFU protocol consist six phases: the user registration phase, the agent registration phase, the renewal phase, the user authentication phase, the synchronous request phase, and the synchronous confirmation phase. The user registration process and the agent registration process are performed only once, and the user authentication phase is executed every time the user is authenticated. The synchronous request process is executed after the user authentication process, and the synchronous confirmation process may be executed when the user wants to confirm the server's stores. The renewal process is executed when a new agent joins the service or the synchronous request process has been successful. These six phases are described below.

3.2.1 User Registration Phase

For use of the service, the user registers her/his identity and the verifier to the server. The user registration process is below.

1. U inputs ID_U and generates N_{U1} and N_{U2}. Next U calculates G_{U1} and G_{U2}. Then U stores G_{U1} and G_{U2} in her/his secret memory such as a smart card.
2. $U \Rightarrow S$: ID_U, G_{U1}, and G_{U2}.
3. S stores G_{U1} and G_{U2} with ID for subsequent authentication. Next S generates N_{S0}, N_{S1} and calculates G_{S0}, G_{S1}, and hG_{U1}, G_{S1}. Then S stores the calculated data.
4. $S \Rightarrow U$: G_{S0} and $h(G_{U1}, G_{S1})$.

3.2.2 Agent Registration Phase

For participation of the service, the agent registers her/his identity and the verifier. The agent registration process is below.

1. A generates N_{A1} and calculates G_{A1}. Then A stores the calculated data.
2. $A \Rightarrow S$: ID_A and G_{A1}.
3. S stores the received data.

3.2.3 Renewal Phase

For user access, the server registers or renews the authentication data to agents. Then A is storing G_{Ai}, and S is storing G_{Ui}, G_{Ui+1}, G_{Si-1}, G_{Si}, G_{Ai}, and $h(G_{Ui}, G_{Si})$. The renewal process is below.

1. S generates N_{S2}. Then S calculates and stores the following data.

$$
\begin{aligned}
M_iU_S &= m(G_{Si-1}, G_{Si}),\\
M_iU_A &= m(G_{Ai}, h(G_{Ui}, G_{Si-1})),\\
M_iU_{US} &= m(h(G_{Ui}, G_{Si}), h(G_{Ui+1}, G_{Si+1})),\\
H_iU_{AS} &= h(G_{Si}, G_{Ai}, h(G_{Ui+1}, G_{Si+1})),\\
H_iA_U &= h(G_{Ui}, G_{Si}, G_{Ai}),\ \textit{and}\\
H_iA_S &= h(G_{Ai}, M_iU_S, M_iU_A, M_iU_{US},\\
&\qquad H_iU_{AS}, H_iA_U).
\end{aligned}
$$

2. $S \Rightarrow A$: M_iU_S, M_iU_A, M_iU_{US}, H_iU_{AS}, H_iA_U, H_iA_S, with ID_U
3. A calculates $h(G_{Ai}, M_iU_S, M_iU_A, M_iU_{US}, H_iU_{AS}, H_iA_U)$ and compares the calculated data and the received H_iA_S. If they are the same, S is authenticated, and A stores the received data.

3.2.4 User Authentication Phase

To log in, U executes the ith user authentication session of the SAIFU protocol. Then U is storing G_{Ui}, G_{Ui+1}, G_{Si-1}, and $h(G_{Ui}, G_{Si})$, and A is storing G_{Ai}, M_iU_S, M_iU_A, M_iU_{US}, H_iU_{AS}, H_iA_U, with ID_U. Figure 2 shows the ith user authentication phase of the SAIFU protocol.

1. $U \rightarrow A$: Service request with ID_U.
2. $A \rightarrow U$: M_iU_S, M_iU_A, M_iU_{US}, and H_iU_{AS}.
3. After receiving A's response, U calculates $h(G_{Ui}, G_{Si-1})$ and retrieves G_{Si}, G_{Ai}, and $h(G_{Ui+1}, G_{Si+1})$ from the received M_iU_S, M_iU_A, M_iU_{US}. Next, U calculates $h(G_{Si}, G_{Ai}, h(G_{Ui+1}, G_{Si+1}))$ and compares the calculated data and the received H_iU_{AS}. If they are the same, A and S are authenticated, and U calculates the following data.

$$
\begin{aligned}
&h(G_{Ui}, G_{Si}, G_{Ai}),\\
M_iS_U &= m(G_{Ui+1}, G_{Ui+2}),\\
H_iS_U &= h(G_{Ui+1}, G_{Ui+2}),\ \textit{and}\\
H_iA_{UD} &= h(G_{Ai}, M_iS_U, H_iS_U).
\end{aligned}
$$

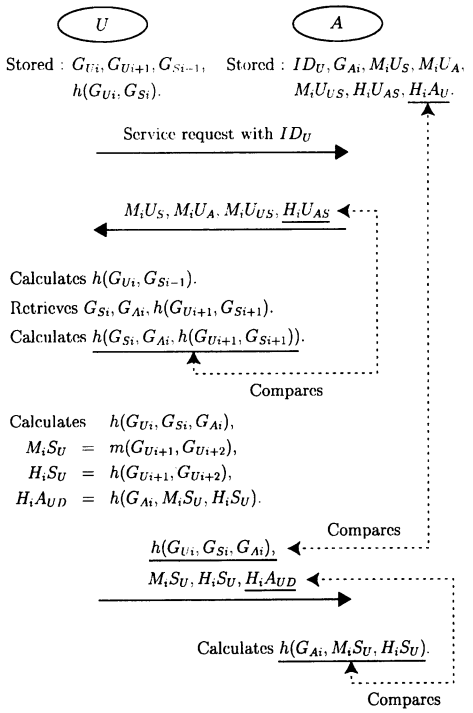

figure 2 The ith user authentication phase of the SAIFU.

4. $U \rightarrow A$: $h(G_{Ui}, G_{Si}, G_{Ai})$, M_iS_U, H_iS_U, and H_iA_{UD}.
5. A compares the storing H_iA_U and the received $h(G_{Ui}, G_{Si}, G_{Ai})$. If they are the same, U is authenticated. Next, A calculates $h(G_{Ai}, M_iS_U, H_iS_U)$, and compares the calculated data and the received H_iA_{UD}. If they are the same, the received data M_iS_U and H_iS_U are not forged, and A stores them.

3.2.5 Synchronous Request Phase

For the next user authentication phase, A renews the next verifiers. Then A storing M_iS_U, H_iS_U, with ID_U, and S is storing G_{Ui}, G_{Ui+1}, G_{Si-1}, G_{Si}, G_{Si+1}, G_{Ai}, $h(G_{Ui}, G_{Si})$, and $h(G_{Ui+1}, G_{Si+1})$. The synchronous request process is below.

1. A generates N_{Ai+1} and calculates G_{Ai+1},

$$M_iS_A = m(G_{Ai}, G_{Ai+1}), \text{ and}$$
$$H_iS_A = h(G_{Ai+1}, M_iS_U, H_iS_U).$$

2. $A \rightarrow S$: M_iS_A, M_iS_U, H_iS_U, H_iS_A, ID_U, with ID_A.
3. First, S retrieves G_{Ai+1} from the received M_iS_A. Next, S calculates $h(G_{Ai+1}, M_iS_U, H_iS_U)$ and compares the calculated data and the received H_iS_A. If they are the same, A is authenticated. Then S retrieves G_{Ui+2} from the received M_iS_U, and authenticates the retrieved data by comparing the calculated $h(G_{Ui+1}, G_{Ui+2})$ and the received H_iS_U. If they are the same, A's and U's verifiers are stored, and S calculates $h(G_{Ai}, G_{Ai+1})$.
4. $S \Rightarrow A$: $h(G_{Ai}, G_{Ai+1})$
5. A compares the calculated $h(G_{Ai}, G_{Ai+1})$ and the received data. If they are the same, S's mutual authentication is succeed.

3.2.6 Synchronous Confirmation Phase

For the next user authentication phase, U can confirm her/his verifiers, which is stored in the server. Then U is storing G_{Ui}, G_{Ui+1}, G_{Ui+2}, G_{Si-1}, G_{Si}, G_{Ai}, $h(G_{Ui}, G_{Si})$, $h(G_{Ui+1}, G_{Si+1})$, M_iS_U, H_iS_U, H_iA_{UD}, and S is storing G_{Ui}(or G_{Ui+2}), G_{Ui+1}, G_{Si-1}, G_{Si}, G_{Si+1}, G_{Ai} (or G_{Ai+1}), $h(G_{Ui}, G_{Si})$, $h(G_{Ui+1}, G_{Si+1})$. The synchronous confirmation process is below.

1. $U \rightarrow S$: M_iS_U, H_iS_U, H_iA_{UD} with ID_U.
2. S checks the receiving data using the stored U's verifiers G_{Ui}, G_{Ui+1} (or G_{Ui+1}, G_{Ui+2}); S computes $h(m(M_iS_U, G_{Ui}), G_{Ui})$ (or $h(m(M_iS_U, G_{Ui+1}), G_{Ui+1})$) and compares the computed data and the received H_iS_U. If they are the same, U's verifiers are renewed, and S calculates $h(G_{Si}, G_{Ai})$.
3. $S \Rightarrow U$: $h(G_{Si}, G_{Ai})$
4. U compares the calculated $h(G_{Si}, G_{Ai})$ and the received data. If they are the same, U's verifier is confirmed, and U removes the stored G_{Ui}, G_{Si-1}, $h(G_{Ui}, G_{Si})$, M_iS_U, and H_iS_U.

4. Security and Performance Analysis

The proposal schemes are secure and are useful to various secret memories.

4.1 Security Considerations

One-time password authentication schemes are vulnerable to certain kinds of attacks [6], [7], [10], [12]. The SAIFU scheme is secure against those attacks.

4.1.1 Guessing Attack

In the SAIFU, all of verifiers are encrypted using a random number, and original data cannot be easily retrieved. Moreover, an attacker cannot steal a random number because the previous verifier is stored in tamper resistant device such

as smart card instead of storing a random number in user's terminal. Therefore, guessing attacks fail.

4.1.2 Replay Attack

Verifiers are changed every time, and communication data are different from the present data. Therefore, an attacker cannot replay the communication data, and the SAIFU eliminates the possibility of a replay attack.

4.1.3 Fogery Attack

In the forgery attacks, an attacker forges communication data and changes the user's verifier. In the SAIFU, the authentication data are generated using the verifier, and those authentication data cannot be generated without the verifier. Then an attacker cannot create authentication data because she/he does not have the verifier, which is changed every time. Thus, SAIFU is secure against forgery attacks.

4.1.4 Denial of Service Attack

One-time password authentication schemes are vulnerable to denial of service attacks, in which an attacker forges communication data and disturbs the user's access by changing the user's verifier. The SAIFU scheme is secure against forgery attacks, and the attacker cannot change the verifier. Moreover, the attacker cannot create the authentication data which is past the server's verifier because all of communication data to authenticate the user are created by using two or more data. Therefore, denial of service attacks fail.

4.2 Performance Considerations

The performance of SAS-2 key-free system [13] and SAIFU scheme is evaluated here. Table 1 summarizes the calculation costs of SAS-2 and SAIFU schemes in the authentication phase.

table 1 Calculation costs of SAS-2 and SAIFU schemes.

	HO_U	HO_A	$I_{U\leftrightarrow A}$	Multi-agent
SAS-2	2	4	3	lack
SAIFU	6	1	3	support

HO_X denotes the number of hashing operations on X.
$I_{X\leftrightarrow Y}$ denotes the number of interactions between X and Y.

These performance evaluations are done in terms of number of hashing operations, number of iterations, and support of multi-agent system. The SAIFU scheme has costs than the SAS-2 key-free system but this isn't fatal. In the SAIFU scheme, the user can be identified by many agents without unpracticed costs.

5. Conclusion

For ubiquitous computing, functional authentication scheme is desirable. Here, we propose a new authentication scheme, in which the user can be identified by many agents. This cannot be executed by existing schemes such as the SAS-2.

References

[1] R. Rivest, "The MD5 message-digest algorithm," Internet Request For Comments 1321, April 1992.

[2] W. Stallings, "Secure hash algorithm," in Cryptography and Network Security: Principles and Practice 2nd ed., pp.193–197, Prentice-Hall, 1999.

[3] L. Lamport, "Password authentication with insecure communication," Commun. ACM, vol.24, no.11, pp.770–772, 1981.

[4] N. Haller, "The S/KEY(TM) one-time password system," Proc. Internet Society Symposium on Network and Distributed System Security, pp.151–158, 1994.

[5] A. Shimizu, "A dynamic password authentication method using one-way function," System and Computers in Japan, vol.22, no.7, pp.32–40, July 1991.

[6] N. Haller and R. Atkinson, "On internet authentication," Internet Request For Comments 1704, Oct. 1994.

[7] C. J. Mitchell and L. Chen, "Comments on the S/Key user authentication scheme," ACM Operating Systems Review, vol.30, no.4, pp.12–16, Oct. 1996.

[8] A. Shimizu, T. Horioka, and H. Inagaki, "A password authentication method for contents communication on the internet," IEICE Trans. Commun., vol.E81-B, no.8, pp.1666–1673, Aug. 1998.

[9] M. Sandirigama, A. Shimizu, and M.-T. Noda, "Simple and secure password authentication protocol (SAS)," IEICE Trans. Commun., vol.E83-B, no.6, pp.1363–1365, June 2000.

[10] C. Lin, H. Sun, and T. Hwang, "Attacks and solutions on strong-password authentication," IEICE Trans. Commun., vol.E84-B, no.9, pp.2622–2627, Sept. 2001.

[11] T.-C. Yeh, H.-Y. Shen, and J.-J. Hwang, "A secure one-time password authentication scheme using smart cards," IEICE Trans. Commun., vol.E85-B, no.11, pp.2515–2518, Nov. 2002.

[12] T. Tsuji and A. Shimizu, "An impersonation attack on one-time password authentication protocol OSPA," IEICE Trans. Commun., vol.E86-B, no.7, pp.2182–2185, July 2003.

[13] T. Tsuji and A. Shimizu, "A one-time password authentication protocol for mobile communications and internet protocols," IEICE Trans. Commun., vol.E87-B, no.6, pp.1594–1600, June 2004.