L-022

One-Time Password Authentication Scheme to Solve Stolen Verifier Problem

Ryoichi Isawa †　Masakatu Morii ‡

1 Introduction

Secure authentication schemes between an authentication server and users are required to avoid many risks on the Internet. There are three authentication schemes: static password authentications like Basic and Digest Access Authentication[1], public-key certificate schemes, and one-time password schemes. In spite of using SSL/TLS, the static password authentications are known as being insecure because majority of people use short and simple passwords. An adversary can easily guess their passwords, for example by Library attack. Public-key certificate schemes provide the necessary security. However, it requires heavy computational costs and is not suitable for low spec mobile devices. In contrast, the computational costs of one-time password schemes are lower than that of public-key certificate schemes. They are generally based on a symmetric encryption function or a one-way hash function. In addition, since one-time password schemes generate a different password at every authentication session, they are more secure than the static password authentications. Thus, one-time password schemes is an important area of research.

There are three types of one-time password scheme: 1) based on time-synchronization between the authentication server and the user[2], 2) using a mathematical algorithm where the new one-time password is based on a challenge[3, 4, 5] and 3) using a mathematical algorithm to generate a new one-time password based on the previous one-time password. We focus on the type 3 in this study, because it changes not only the one-time password but also the secret data stored by the server and the user at every authentication session. We can use our biometric information as the user's first secret data without care.

Most schemes[6, 7, 8, 9] are vulnerable to Hybrid Theft attack. An adversary can steal the secret data from an authentication server or can obtain the communication data in Hybrid Theft attack. The adversary tries to impersonate a legal user using the stolen data. Since the adversary can obtain the server's secret data, the Hybrid Theft attack is very difficult to immunize. 2GR[10] is designed to immunize the Hybrid Theft attack, but it suffers from the Impersonation attack[11]. SAS-X(2)[12] is secure against the Hybrid Theft attack, but suffers from DoS attack.

We classify one-time password schemes into two designs: 1) based on only a one-way hash function and 2) combining a symmetric encryption function and a one-way hash function. The former is superior in computational cost to the latter, because one-way hash functions can compute faster than symmetric encryption functions. In addition, the latter uses two functions. If we make the circuit board of a one-time password scheme, the area resources of the former become smaller than that of the latter. On the other hand, we could easily make a secure scheme of the latter. The schemes[6, 7, 8, 9, 10] are the former, and SAS-X(2)[12] is the latter.

In this paper, we propose a new one-time password scheme against the Hybrid Theft attack[*1]. The proposed scheme has three advantages: 1) secure against all the existing attacks, 2) based on only one-way hash function, and 3) a mutual authentication scheme. Compared with SAS-X(2), the proposed scheme is more secure because of the advantage 1, and can compute faster because of the advantage 2. SAS-X(2) is a one-way authentication scheme. In contrast, the proposed scheme is a mutual authentication scheme.

2 Attacks on One-Time Password Scheme

There are ten attacks on one-time password scheme. The Replay attack, the Forgery attack, the Impersonation attack, and the DoS attack are defined in [7]. The SV attack is defined in [15]. The Theft attack and the Server Modification attack are defined in [10]. The SV DoS attack is defined in [14]. In this paper, we call the SV DoS attack Theft DoS attack because of the same assumption of the SV DoS attack[14]. We define the Hybrid Theft attack and the Server Impersonation attack in this section.

Replay attack: An adversary obtains the communication data between the server and the user in the past authentication sessions. In the current authentication session, she/he replaces all or a specific part of the communication data with the obtained data. If it succeeds, she/he impersonates a legal user from the next authentication session.

Forgery attack: An adversary modifies the communication data in the current authentication session. If it succeeds, she/he impersonates a legal user from the next authentication session.

Impersonation attack: An adversary uses both the Replay attack and the Forgery attack in order to impersonate a legal user.

Server Impersonation attack: An adversary uses the Impersonation attack in order to impersonate the legal server of a specific user. One-way authentication schemes are vulnerable to this attack.

Denial of Service attack (DoS attack): An adversary

† Graduate School of Science and Technology, Kobe University
‡ Graduate School of Engineering, Kobe University

[*1] In 2007, we proposed the scheme against the Hybrid Theft attack[13]. In this paper, we improve the scheme[13] for the server and the user to update all secret data at every authentication session. In [14], Nakayama et al. claimed the scheme[13] suffers from the Theft DoS attack. But it is not true. We omit that proof in this paper.

uses the Replay attack or the Forgery attack in order to alter the server's secret data or the user's one. If it succeeds, the server and the user cannot authenticate each other from the next authentication session.

Stolen Verifier attack (SV attack): An adversary steals verification data from the server in the current or past authentication sessions. Here, the verification data does not include secret keys used with XOR operation or an encryption function. She/he generates communication data using the stolen data and sends them to the server. If it succeeds, she/he impersonates a legal user from the next authentication session.

Theft attack: An adversary steals the secret data from the server in the current or past authentication sessions. Here, the secret data mean both the verification data and the secret keys. She/he generates communication data using the stolen data and sends them to the server. If it succeeds, she/he impersonates a legal user from the next authentication session.

Hybrid Theft attack: An adversary uses both the Theft attack and the Impersonation attack in order to impersonate a legal user.

Theft DoS attack: An adversary uses both the Hybrid Theft attack in order to alter the server's secret data or the user's one. If it succeeds, the server and the user cannot authenticate each other from the next authentication session.

Server Modification attack: An adversary directly modifies the server's secret data. Tsuji et al. says that this attack is more unrealistic than the Theft attack, because the 'writing' is under high restriction than the 'reading' in most of systems[10]. We also do not consider this attack.

3 Weaknesses of Existing Schemes

In 2002, Tsuji et al. proposed SAS-2 (Simple And Secure password authentication protocol, ver.2), which reduced the times of a one-way hash function operated at the server and the user[6, 7]. SAS-2 was superior in computational cost to the revised SAS[16] with the equivalent security level. In this scheme, Tsuji et al. assumed any adversaries cannot steal the server's secret data. Naturally, SAS-2 is vulnerable to the Hybrid Theft attack.

Chien et al. proposed the ROSI (RObust and SImple authentication protocol), which immunizes the SV attack[8]. Chien et al. assumed the server had the secret key which an adversary could not steal. Tsuji et al. pointed out that ROSI was vulnerable to the Hybrid Theft attack and proposed 2GR (Two-Gene-Relation password authentication protocol)[10] in order to immunize the Hybrid Theft attack. However, Lin et al. found that 2GR was vulnerable to the Impersonation attack[11]. In the Impersonation attack, an adversary tries to impersonate a legal user using the intercepted communication data. There are two reasons why 2GR was vulnerable to the attack: 1) 2GR is a one-way authentication scheme (user to server) and 2) the server updates the new verifier without any integrity check. Kuo et al. also indicated that 2GR was vulnerable to the Impersonation attack and proposed an improved scheme immunizing the Hybrid Theft attack[9]. Unfortunately, Kim et al. pointed out that Kuo et al.'s scheme was still vulnerable to the attack[17]. Although Kim et al. proposed a new scheme, they did not prove that their scheme was secure against the Theft attack. In order to immunize the Hybrid Theft attack, Tsuji et al. proposed SAS-X(2)[12]. However, it suffers from the Denial of Service attack (DoS attack)[18]. Thus all the existing schemes have certain vulnerabilities.

Table 1 List of symbols used in this paper

U	a user who requests the server to authenticate her/himself
S	the authentication server
A	an adversary
ID	a user's identification
PW	a user's password
$h(x)$	the hash value of the input data x
$\check{x}$	used when U or S compares certain data with $\check{x}$
x'	used when A chooses x' at random instead of a valid x, or used when A calculates certain data y' from x'
X⇒Y: Z	X sends Z to Y through a secure channel
X→Y: Z	X sends Z to Y through an insecure channel
$\|\|$	a concatenation
$\oplus$	XOR operation

4 Proposed Scheme

Table 1 shows a list of symbols used in this paper. The proposed scheme consists of two phases: the registration phase and the authentication phase. In order to immunize the Hybrid Theft attack, U does not send a next verifier A_{i+1} to S and S does not store it in the ith authentication session. Since U creates A_{i+1} using a random number Q_i in the $(i-1)$th authentication session, U will never send Q_i to S and S does not store it.

4.1 Registration Phase

1. U inputs ID and PW.
2. U⇒S: ID.
3. S generates three random numbers R_0, R_{-1}, and F_0.
4. S⇒U: R_0, R_{-1}, F_0.
5. U calculates the following data.

$$A_1 = h(ID||PW||F_0)$$
$$F_1 = h(A_1)$$
$$Q_1 = h(PW||R_{-1})$$
$$A_2 = h(ID||Q_1||F_1)$$
$$F_2 = h(A_2)$$
$$Q_2 = h(Q_1||R_0)$$

$$V_1 = h(A_1||F_2)$$

6. U stores ID, Q_2, A_1, F_1, A_2, and F_2.
7. U⇒S: F_1, V_1.
8. S stores ID, F_1, and V_1.

4.2 Authentication Phase

Let us illustrate the protocol as follows. U has stored ID, Q_{i+1}, A_i, F_i, A_{i+1}, and F_{i+1}. S has stored ID, F_i, and V_i.

1. U calculates the following data.

$$A_{i+2} = h(ID||Q_{i+1}||F_{i+1})$$
$$F_{i+2} = h(A_{i+2})$$
$$F_{i+1} \oplus F_i$$
$$A_i \oplus F_{i+1}$$
$$V_{i+1} = h(A_{i+1}||F_{i+2})$$
$$h(F_i||V_{i+1})$$

2. U→S: ID, $F_{i+1} \oplus F_i$, $A_i \oplus F_{i+1}$, V_{i+1}, $h(F_i||V_{i+1})$.
3. **S obtains F_{i+1} and A_i, and verifies their validity as follows.** S extracts F_{i+1} from the received $F_{i+1} \oplus F_i$ using the stored F_i, and extracts A_i from the received $A_i \oplus F_{i+1}$ using the obtained F_{i+1}. S calculates $\check{F}_i = h(A_i)$ using the obtained A_i, and compares $\check{F}_i$ with the stored F_i. If they match, S is convinced that the received F_{i+1}, F_i, and A_i have not been modified; otherwise S terminates this authentication session.
4. **S tries to authenticate U using V_i as follows.** S calculates $\check{V}_i = h(A_i||F_{i+1})$, and compares $\check{V}_i$ with the stored V_i. If they match, S authenticates U; otherwise, S detects the Theft DoS attack and terminates this authentication session.
5. **S verifies the validity of V_{i+1} as follows.** S calculates $h(F_i||V_{i+1})$ using the stored F_i and the received V_{i+1}. Then S compares it with the received $h(F_i||V_{i+1})$. If they match, S is convinced that V_{i+1} have not been modified; otherwise S terminates this authentication session.
6. S calculates $h(R_i||F_i)$ using a randomly selected number R_i, and stores F_{i+1} and V_{i+1}.
7. S→U: R_i, $h(R_i||F_i)$.
8. **U tries to authenticate S as follows.** U calculates $h(R_i||F_i)$ using the received R_i and the stored F_i. Then U compares the calculated $h(R_i||F_i)$ with the received one. If they match, U authenticates S; otherwise, U terminates this authentication session.
9. U calculates $Q_{i+2} = h(Q_{i+1}||R_i)$ and stores Q_{i+2}, A_{i+2}, and F_{i+2}.

4.3 Security Analysis

This section discusses the security of the proposed scheme against various attacks. The security analysis on the Hybrid Theft attack shows in Sect.4.4.

Replay attack: A uses the past communication data in the Replay attack. If U changes all the communication data at random at every authentication session, we conclude the one-time password scheme is secure against the Replay attack. In the proposed scheme, all the communication data except for ID are $F_{i+1} \oplus F_i$, $A_i \oplus F_{i+1}$, V_{i+1}, and $h(F_i||V_{i+1})$ in the ith authentication session. U updates A_i to A_{i+1} using a random number Q_i at every authentication session. Similarly, U updates F_i to F_{i+1} using such A_i. Since $F_{i+1} \oplus F_i$, $A_i \oplus F_{i+1}$, and $h(F_i||V_{i+1})$ include F_i or A_i, they changes at every authentication session. Likewise, U updates A_{i+1} to A_{i+2} using a random number Q_{i+1} so that $V_{i+1}(= h(A_{i+1}||F_{i+2}))$ changes at every authentication session. Therefore the proposed scheme is secure against from the Replay attack.

Forgery attack: A modifies the communication data in the ith authentication session. If S verifies whether or not all the communication data are modified, we conclude the one-time password scheme is secure against the Forgery attack. In the proposed scheme, S verifies $F_{i+1} \oplus F_i$ and $A_i \oplus F_{i+1}$ at step 3 in the authentication phase. At step 5, S verifies V_{i+1} and $h(F_i||V_{i+1})$. Therefore the proposed scheme is secure against the Forgery attack.

Impersonation attack: In the proposed scheme, U changes all the communication data at random at every authentication session. S also verifies them. We conclude that the proposed scheme is secure against the Impersonation attack.

Server Impersonation attack: This security analysis is similar to that of the Impersonation attack. In the proposed scheme, S changes R_i and $h(R_i||F_i)$ at random at every authentication session. U also verifies them at step 8 in the authentication phase. Therefore the proposed scheme is secure against the Server Impersonation attack.

DoS attack: A has to modify the communication data in order to alter S's secret data or U's secret data. In the proposed scheme, both S and U verify all the communication data. Therefore the proposed scheme is secure against the DoS attack.

Theft attack: In the proposed scheme, S authenticates U using the received A_i and F_{i+1}. S does not store them and the data from which A_i and F_{i+1} are created. Since A cannot obtain A_i and F_{i+1}, the proposed scheme is secure against the Theft attack.

Theft DoS attack: Suppose that A has stolen F_i from S. A modifies V_{i+1} and $h(F_i||V_{i+1})$ as V'_{i+1} and $h(F_i||V'_{i+1})$ at random respectively. S updates V_i to the received V'_{i+1}. In the $(i+1)$th authentication session, S can detect the Theft DoS attack at step 4 in the authentication phase. Therefore the proposed scheme is secure against the Theft DoS attack. However, since S cannot authenticate U, S terminates the authentication session. Here, let us give a variation for S to continue the authentication as follows. Suppose that S and U have the current secret data and the previous one. When S detects the Theft DoS attack, S and U use the previous secret data. Having authenticated each other, they updates own previous secret data to the current secret data. The variation on the proposed scheme can immunize the Theft DoS attack completely.

4.4 Security Analysis on Hybrid Theft attack

In one-time password schemes, A can certainly impersonate a legal user only once in the *i*th authentication session as follows. A intercepts the communication data and directly sends them to S. Since S receives the valid data, S is convinced that A is the legal user. Similarly, in the Hybrid Theft attack, A can certainly impersonate a legal user only once. The question we should consider is whether A can impersonate U in the $(i + 1)$th authentication session.

In the proposed scheme, S verifies $F_{i+1} = h(A_{i+1})$ at step 3 in the $(i + 1)$th authentication session. In order to impersonate U, 1) A has to obtain a valid A_{i+1} or 2) A has to make S store $F'_{i+1} = h(A'_{i+1})$ in the *i*th authentication session. Suppose that A steals F_i from S. Then A intercepts the communication data, and extracts A_i and F_{i+1} from the intercepted data using the stolen F_i. Now A has U's secret data except for A_{i+1} and Q_{i+1} in the *i*th authentication session. However, since A cannot obtain A_{i+1} $(= h(ID||Q_i||F_i))$, Q_{i+1}, and also Q_i, A cannot take the method 1. On the other hand, even if A replaces $F_{i+1} \oplus F_i$ and $A_i \oplus F_{i+1}$ with $F'_{i+1} \oplus F_i$ and $A_i \oplus F'_{i+1}$ respectively on the network, S rejects the requests of A at step 4 in the *i*th authentication session. Thus, A cannot take the method 2. Therefore the proposed scheme is secure against the Hybrid Theft attack.

5 Conclusion

The existing schemes are vulnerable to some attacks. In this paper, we proposed a new one-time password scheme against the Hybrid Theft attack. Moreover, only the proposed scheme is secure against all the existing attacks. In addition, the proposed scheme can compute faster because it is based on only a one-way hash function. For the same reason, when we make the circuit board of the proposed scheme, the area resources become smaller. We can apply the proposed scheme to low spec devices and design a secure authentication system.

Reference

[1] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart, "Http authentication: Basic and digest access authentication," Internet Request For Comments 2617, June 1999.

[2] N. Sklavos and C. Efstathiou, "Securid authenticator: On the hardware implementation efficiency," proceedings of 14th IEEE International Conference on Electronics, pp.589–592, Dec. 2007.

[3] M. Peyravian and N. Zunic, "Methods for protecting password transmission," Computers and Security, vol.19, no.5, pp.466–469, July 2000.

[4] T.H. Chen and W.B. Lee, "A new method for using hash functions to solve remote user authentication," Computers and Electrical Engineering, vol.34, pp.53–62, Jan. 2008.

[5] J.Y. Kim, H.K. Choi, and J.A. Copeland, "Further improved remote user authentication scheme," IEICE Trans. Fundamentals, vol.E94-A, no.6, pp.1426–1433, June 2011.

[6] T. Tsuji and A. Shimizu, "Simple and secure password authentication protocol, ver.2 (sas-2)," IEICE Technical Report, OIS2002-30, vol.46, no.2, pp.7–11, Sept. 2002.

[7] T. Tsuji and A. Shimizu, "A one-time password authentication method for low spec machines and on internet protocols," IEICE Trans. Commun., vol.E87-B, no.6, pp.1594–1600, June 2004.

[8] H.Y. Chien and J.K. Jan, "Robust and simple authentication protocol," Comput. J., vol.46, no.2, pp.193–201, Feb. 2003.

[9] W.C. Kuo and Y.C. Lee, "Attack and improvement on the one-time password authentication protocol against theft attacks," Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, pp.1918–1922, Aug. 2007.

[10] T. Tsuji and A. Shimizu, "One-time password authentication protocol against theft attacks," IEICE Trans. Commun., vol.E87-B, no.3, pp.523–529, March 2004.

[11] C.L. Lin and C.P. Hung, "Impersonation attack on two-gene-relation password authentication protocol (2gr)," IEICE Trans. Commun., vol.E89-B, no.12, pp.3425–3427, Dec. 2006.

[12] T. Tsuji, T. Nakahara, and A. Shimizu, "A one-time password authentication method," IEICE Technical Report, OIS2005-83, vol.111, no.1, pp.23–28, Jan. 2006.

[13] S. Nakano, H. Kuwakado, and M. Morii, "Analysis of secure mutual authentication methods against the stolen-verifier attack," IEICE Technical Report, OIS2007-37 (Japanese Edition), vol.107, no.230, pp.61–64, Sept. 2007.

[14] Y. Nakayama, T. Tsuji, and A. Shimizu, "A one-time password authentication scheme resistant to dos attacks," IEICE Technical Report, OIS2008-87, vol.111, no.1, pp.23–28, Jan. 2009.

[15] C.M. Chen and W.C. Ku, "Stolen-verifier attack on two new strong-password authentication protocols," IEICE Trans. Commun., vol.E85-B, no.11, pp.2519–1521, Nov. 2002.

[16] T. Kamioka and A. Shimizu, "The examination of the security of sas one-time password authentication," IEICE Technical Report, OFS2001-48, pp.53–58, Nov. 2001.

[17] M. Kim, B. Lee, S. Kim, and D. Won, "Weaknesses and improvements of a one-time password authentication scheme," International Journal of Future Generation Communication and Networking, vol.2, no.4, pp.29–38, Dec. 2009.

[18] K. Uo, Y. Shiraishi, and M. Morii, "Evaluation of a one-time password method to resist stolen-verifier attack," Proceedings of Computer Security Symposium 2006 (Japanese Edition), CD-ROM, pp. 8A-4, 6 pages, Oct. 2006.