A Common Weakness of Password Authentication Schemes Requiring Synchronous Update of Stored Data

Wei-Chi Ku
Department of Computer Science and Information Engineering
Fu Jen Catholic University
Email: wcku@csie.fju.edu.tw

Hao-Chuan Tsai
Department of Computer Science and Information Engineering
Fu Jen Catholic University
Email: saul91@csie.fju.edu.tw

Maw-Jinn Tsaur
Graduate Institute of Applied Science and Engineering
Fu Jen Catholic University
Email: mtsao@dns.ee.tnit.edu.tw

Abstract- *To resist off-line password guessing attacks without using public-key techniques, many newer hash-based one-time password authentication schemes additionally employ a smart card to generate a one-time high-entropy passcode from the user-chosen fixed password. Among these schemes, some require synchronous update of stored data in the user's smart card and the server. Herein, we show that such schemes tend to suffer from denial-of-service attacks by using three illustrative examples.*

Keywords: password authentication, data synchronization, smart card, denial-of-service attack.

1. Introduction

Password authentication is widely used for its simplicity, convenience, adaptability, and mobility. Traditional static password authentication schemes are vulnerable to eavesdropping attacks in open network environments, and thus cannot meet nowadays security requirements. To solve this problem, many one-time password authentication schemes have been proposed. Roughly, these schemes can be categorized into two types [1][4][5][7][9][11][15][17][19][20], the ones mainly using public-key techniques and the other ones mainly using hash functions. However, the former type, e.g., [4][1][19][7][9][6], usually involves complicated computations, and therefore is unsuitable for some constrained environments. In contrast, the latter type, which is the focus of this paper, has the advantage of lighter computational overhead.

In 1981, Lamport [10] initially described a one-time password authentication scheme based on hash functions. Lamport's scheme allows the server to authenticate the user in a way that neither eavesdropping on an authentication exchange nor reading server's database can enable the adversary to impersonate the user. Based on Lamport's scheme, Haller [5] derived a one-time password scheme, S/KEY, which can be used to control user access to remote servers. However, S/KEY was found to be vulnerable to a server spoofing attack and a replay attack [14]. Independently, Shimizu [16] proposed a one-time password authentication scheme, CINON, in which the user has to memorize two variable random numbers. These inconveniences obstruct the deployment of CINON. To improve CINON, Shimizu, Horioka, and Inagaki [17] proposed a one-time password authentication scheme, PERM, in which a sequential number is stored in the server for authenticating the user. Later, PERM was found to be vulnerable to a man-in-the-middle attack in that the adversary can impersonate the user by modifying two consecutive sessions between the user and the server. In 2000, Sandirigama, Shimizu, and Noda [15] proposed a simple strong-password authentication scheme SAS, which was intended to be superior to S/KEY [10], CINON [16], and PERM [17] in storage utilization, processing time, and transmission overhead. However, Lin, Sun, and Hwang [11] showed that SAS is vulnerable to a replay attack and a denial-of-service attack, and then proposed a new scheme, OSPA (Optimal Strong-Password Authentication). Unfortunately, Cheng and Ku [3] have found that OSPA cannot effectively resist a stolen-verifier attack. Moreover, OSPA cannot resist a man-in-the-middle attack [18].

Unfortunately, all above mentioned hash-based one-time password authentication schemes fail to resist the off-line password guessing attack. To resist off-line password guessing attacks without using public-key techniques, some hash-based one-time password authentication schemes additionally employ a smart card to generate a one-time high-entropy passcode from the user-chosen fixed password, e.g., [2][8][12]. The user sends the generated passcode to the server for authentication. Among these hash-based one-time password authentication schemes using smart cards, some require synchronously updating the data stored in both the user's smart card and the server to enhance security. However, we find that these hash-based one-time password authentication schemes requiring synchronous update of stored data have a common weakness in practice, the vulnerability of suffering from various denial-of-service attacks. In this paper, we will illustrate our observation by mounting various denial-of-service attacks on three such schemes of different types, ROSI [2], SAS-1 [8], and the LSH scheme [12]. All the illustrative schemes involve two phases, the registration phase and the authentication phase. The registration phase is invoked only once for registering each user. The authentication phase is invoked whenever a user uses his password to access the resources at the server. In the illustrative schemes, both the data stored in the user and the server should be synchronously updated in the authentication phase.

The notations used throughout this paper can be described as in the following. C represents the user and S represents the server. ID and pw denote the identity and password of C, respectively. Notation N_i denotes the random number generated by C in his $(i-1)$th authentication phase and will be used in his ith authentication phase. $H(\ )$ denotes a cryptographic hash function and x denotes S's secret key. Notations $\oplus$ and $\|$ represent the bitwise XOR and the concatenation operator, respectively.

2. Example I: Weakness of ROSI

In 2003, Chien and Jan [2] proposed a hash-based one-time password authentication scheme, the Robust and Simple authentication protocol (ROSI), which assumes the use of a smart card. ROSI allows the user to freely choose his easily memorized password and store a strong secret key in his smart card. They claimed that ROSI can resist the replay attack, the impersonation attack, the man-in-middle attack, the stolen-verifier attack, the off-line password guessing attack, and the masqueraded server attack. They also claimed that ROSI can achieve robust security with lower transmission cost. Next, we will show that ROSI is vulnerable to a denial-of-service attack in that the user will be fooled into abandoning updating the stored data in his smart card while the server has updated his stored data.

2.1. Scheme Description

The registration phase and the authentication phase of ROSI can be briefly described as in the following.

The registration phase of ROSI

C sends ID, pw, and a random number N_1 to S through a secure channel. S computes $H(pw\|N_1)$ and $H^2(pw\|N_1)$, stores ID and $H^2(pw\|N_1)$ as the initial verifier of C's password pw, and then issues a smart card containing R $(= H(x\|ID) \oplus pw)$ and $H(pw\|N_1)$ to C.

The *i*th authentication phase of ROSI

Step 1. $C \rightarrow S$: ID, c_1, c_2.

where

$$c_1 = H(H(x\|ID) \oplus H^2(pw\|N_i)) \oplus H^2(pw\|N_{i+1})$$

$$c_2 = H^3(pw\|N_{i+1}) \oplus H(pw\|N_i)).$$

Step 2. $C \leftarrow S$: $H^3(pw\|N_{i+1}) \oplus H^2(pw\|N_i)$.

C keys in pw to his smart card, which will then generate a random number N_{i+1} and use pw to extract $H(x\|ID)$ from the stored R. Next, C's smart card uses pw, N_{i+1}, the extracted $H(x\|ID)$, and the stored $H(pw\|N_i)$ to compute the passcode $\{c_1, c_2\}$, which is then sent to S along with ID. Upon receiving C's passcode, S computes $H(H(x\|ID) \oplus H^2(pw\|N_i))$ by using his secret key x and the stored $H^2(pw\|N_i)$, and then uses the computed result to extract $H^2(pw\|N_{i+1})$ from the received c_1. Next, S applies $H(\ )$ to the extracted $H^2(pw\|N_{i+1})$ and uses the result to extract $H(pw\|N_i)$ from the received c_2. Subsequently, S applies $H(\ )$ to the extracted $H(pw\|N_i)$ and checks whether the result equals the stored $H^2(pw\|N_i)$. If it holds, S accepts C's login request, updates the stored $H^2(pw\|N_i)$ with $H^2(pw\|N_{i+1})$, and sends $H^3(pw\|N_{i+1}) \oplus H^2(pw\|N_i)$ to C. If the received message equals $H^3(pw\|N_{i+1}) \oplus H^2(pw\|N_i)$, which can be computed in advance, C's smart card will update the stored $H(pw\|N_i)$ with $H(pw\|N_{i+1})$.

2.2. Denial-of-Service Attack

Next, we show that ROSI is vulnerable to a denial-of-service attack by using the following scenario. During C's ith login, $\{ID, c_1, c_2\}$ is sent to S in Step 1. After successfully verifying the received $\{ID, c_1, c_2\}$, S will replace the stored $H^2(pw\|N_i)$ with $H^2(pw\|N_{i+1})$, and then send $H^3(pw\|N_{i+1}) \oplus H^2(pw\|N_i)$ to C in Step 2. In the meanwhile, the adversary can replace the transmitting $H^3(pw\|N_{i+1}) \oplus H^2(pw\|N_i)$ with an arbitrary equal-sized string. Since the modified message received in Step 2 does not equal the expected one, C's smart card will not update the stored $H(pw\|N_i)$ with $H(pw\|N_{i+1})$. As the data stored in S and C's smart card are inconsistent, C's succeeding login request to S will be denied unless he re-registers to S again.

3. Example II: Weakness of SAS-1

In 2000, Sandirigama, Shimizu, and Noda [15] proposed a hash-based one-time password authentication scheme, the Simple And Secure authentication protocol (SAS), which was aimed to withstand the man in-the-middle attack that can break PERM [17]. Not requiring memory for storing random numbers on the user's side, SAS requires no smart card support. Moreover, SAS requires no resetting of passwords and has low computation and communication costs, which make it more attractive than others [2]. However, SAS was found to be vulnerable to a denial-of-service attack [11] and a stolen-verifier attack [3]. In 2001, Kamioka and Shimizu [8] proposed an improved version of SAS, SAS-1, which requires the support of a smart card. Next, we will show hat SAS-1 is still vulnerable to a denial-of-service attack in that the user updates the stored data in his smart card while the data stored in the server has not been updated.

3.1. Scheme Description

The registration phase and the authentication phase of SAS-1 can be briefly described as in the following.

The registration phase of SAS-1

C sends *ID*, *pw*, and a random number N_1 to *S* through a secure channel. *S* computes $H^2(pw\|N_1)$, stores *ID* and $H^2(pw\|N_1)$ as the initial verifier of *C*'s password *pw*, and then issues a smart card containing N_1 to *C*.

The *i*th authentication phase of SAS-1

Step1. $C \rightarrow S$: ID, c_1, c_2.

where

$$c_1 = H(pw\|N_i) \oplus H^3(pw\|N_{i+1})$$

$$c_2 = H^2(pw\|N_{i+1}) \oplus H^2(pw\|N_i).$$

C keys in *pw* to his smart card, which will then generate the passcode $\{c_1, c_2\}$ by using the stored N_i and the newly generated N_{i+1}, and update the stored N_i with N_{i+1}. Next, *C* sends his passcode along with *ID* to *S*. Then, *S* uses the stored verifier $H^2(pw\|N_i)$ to extract $H^2(pw\|N_{i+1})$ from the received c_2, applies $H(\ )$ to the extracted $H^2(pw\|N_{i+1})$, and uses the result to extract $H(pw\|N_i)$ from the received c_1. Next, *S* applies $H(\ )$ to the extracted $H(pw\|N_i)$ and checks whether the result equals the stored $H^2(pw\|N_i)$. If it holds, *S* updates the stored $H^2(pw\|N_i)$ with $H^2(pw\|N_{i+1})$.

3.2. Denial-of-Service Attack

Next, we will show that SAS-1 is also vulnerable to a denial-of-service attack. During *C*'s *i*th login, the adversary can replace the transmitting passcode with an arbitrary equal-sized string. Since *S* can not derive the correct $H^2(pw\|N_{i+1})$ and $H(pw\|N_i)$ from the received passcode by using the stored $H^2(pw\|N_i)$, he will not update the stored $H^2(pw\|N_i)$ with $H^2(pw\|N_{i+1})$. As N_i has been already replaced by N_{i+1} in *C*'s smart card, *C*'s succeeding login request to *S* will be denied unless he re-registers to *S* again.

4. Example III: Weakness of the LSH Scheme

In 2001, Lin, Sun, and Hwang [11] proposed a hash-based one-time password authentication scheme, the Optimal Strong-Password Authentication protocol (OSPA). However, Chen and Ku [3] pointed out that OSPA is vulnerable to a stolen-verifier attack. Next, Lin, Shen, and Hwang [12] proposed an improved version of OSPA. They claimed that their scheme, denoted by the LSH scheme for short, can resist the off-line password guessing attack, the replay attack, the impersonation attack, and the stolen verifier attack. Next, we will show that the LSH scheme is still vulnerable to a denial-of-service attack in that the adversary can fool the server into updating the stored data with the one that is inconsistent with the updated data of the user's smart card.

4.1. Scheme Description

The registration phase and the authentication phase of the LSH scheme can be briefly described as in the following.

The registration phase of the LSH scheme

C uses *pw* and a random number N_1 to compute $H^2(pw\|N_1)$ and sends the result along with *ID* to *S* through a secure channel. Then, *S* stores $H^2(pw\|N_1)$ as the initial verifier of *C*'s password *pw* and issues a smart card containing K $(= H(x\|ID) \oplus H^2(pw\|N_1))$ and N_1 to *C* through a secure channel.

The *i*th authentication phase of the LSH scheme

Step 1. $C \rightarrow S$: ID, c_2, c_3.

where

$$c_1 = K \oplus H^2(pw\|N_i)$$

$$c_2 = c_1 \oplus H(pw \oplus N_i)$$

$$c_3 = H(c_1) \oplus H^2(pw \oplus N_{i+1}).$$

C keys in *pw* to his smart card, which will then compute c_1, c_2, and c_3 by using the stored N_i and the newly generated N_{i+1}, and update the stored N_i with N_{i+1}. Next, *C* sends his passcode $\{c_2, c_3\}$ along with *ID* to *S*. Then, *S* uses his secret key *x* to compute $H(x\|ID)$ to extract $H(pw \oplus N_i)$ from the received c_2. Next, *S* applies $H(\ )$ to the extracted $H(pw \oplus N_i)$ and checks whether the result equals the stored $H^2(pw||N_i)$. If it holds, S grants C's login request and extracts $H^2(pw \oplus N_{i+1})$ from the received c_3 to replace the stored $H^2(pw \oplus N_i)$ for C's next login.

4.2. Denial-of-Service Attack

Again, we find that the LSH scheme is also vulnerable to a denial-of-service attack. During C's ith login, the adversary can replace the transmitting c_3 with an arbitrary equal-size string, say r. Upon receiving the modified message, S will compute $H(x||ID)$ to extract $H(pw \oplus N_i)$ from the received c_2. Next, S applies $H(\ )$ to the extracted $H(pw \oplus N_i)$. Since the result equals the stored $H^2(pw \oplus N_i)$, S will grant C's login request and update the stored $H^2(pw \oplus N_i)$ with $H^2(x||ID) \oplus r$ instead of $H^2(pw \oplus N_{i+1})$. Clearly, C's succeeding login requests will be denied unless he re-registers to S again.

5. Conclusion

We have shown that three new password authentication schemes requiring synchronous update of stored data in the user's smart card and the server, ROSI, SAS-1, and the LSH scheme, are vulnerable to denial-of-service attacks in different ways. As described, such weaknesses are due to the inconsistence of stored data in the user's smart card and the server. And, it deserves further researches to eliminate such weaknesses without incurring much computation and transmission overhead.

Acknowledgment

This work was partly supported by the National Science Council, R.O.C., under Grant NSC-93-2213-E-030-017.

References

[1] S. Bellovin and M. Merritt, "Encrypted key exchange: password-based protocols secure against dictionary attacks," *Proc. IEEE Symposium on Research in Security and Privacy*, pp.72–84, 1992.

[2] H.Y. Chien and J.K. Jan, "Robust and simple authentication protocol," *The Computer Journal*, vol.46, no.2, pp.193–201, 2003

[3] 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. 2519–2521, Nov. 2002.

[4] W. Diffie, P.C. Van Oorschot, and M.J. Wiener, "Authentication and authenticated key changes," *Designs Codes and Cryptography*, vol.2, no.2, pp.107–125, June 1992.

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

[6] IEEE P1363.2/D12 (Standard specifications for password-based public-key cryptographic techniques), *IEEE P1363 working group*, Dec. 2003.

[7] D. Jablon, "B-SPEKE," *Integrity Sciences white paper*, Sept. 1999.

[8] T. Kamioka and A. Shimizu, "The examination of the security of SAS one-time password authentication," *IEICE Technical Report OFS2001-48,* 2001.

[9] T. Kwon, "Authentication and key agreement via memorable password," *Proc. NDSS 2001 Symposium Conference*, Feb. 2001.

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

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

[12] C.W. Lin, J.J. Shen, and M.S. Hwang: "Security enhancement for optimal strong password authentication protocol," *ACM Operating Systems Review,* vol.37, no.2, pp.7–12, 2003.

[13] C. Mitchell, "Limitations of challenge-response entity authentication," *Electronics Letters*, vol. 25, no. 17, pp. 1195−1196, Aug. 1989.

[14] C.J. Mitchell and L. Chen, "Comments on the S/KEY user authentication scheme," *ACM Operating Systems Review*, vol.39, no.4, pp.12–16, Oct. 1996.

[15] 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.

[16] A. Shimizu, "A dynamic password authentication method by one-way function," *IEICE Trans.*, vol. J73-D-I, no.7, pp.630−636, July 1990.

[17] 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.

[18] 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.

[19] T. Wu, "The secure remote password protocol," *Proc. Internet Society Symposium on Network and Distributed System Security*, 1998.

[20] 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.