Replay and Denial-of-Service Attacks on a New Strong-Password Authentication Scheme

Wei-Chi Ku Hao-Chuan Tsai Shuai-Min Chen

Department of Computer Science and Information Engineering
Fu Jen Catholic University
510 Chung Cheng Rd., Hsinchuang, Taipei County, Taiwan 242, R.O.C.
E-mail: wcku@csie.fju.edu.tw

Abstract

Existing one-time password authentication schemes can be categorized into two types, weak-password authentication schemes and strong-password authentication schemes. Generally, the strong-password authentication schemes have the advantages over the weak-password authentication schemes in that their computational overhead are lighter, designs are simpler, and implementations are easier, and therefore are especially suitable for some constrained environments. Recently, Lin, Sun, and Hwang proposed a strong-password authentication scheme, OSPA, which was later found to be vulnerable to a stolen-verifier attack and a man-in-the-middle attack. Later, Lin, Shen, and Hwang proposed an improved version of OSPA and showed that the improved scheme can resist the guessing attack, the replay attack, the impersonation attack, and the stolen-verifier attack. Herein, we show that their scheme is still vulnerable to a replay attack and a denial-of-service attack.

Keywords: Strong password, password authentication, stolen-verifier attack, replay attack, denial-of-service attack.

1. Introduction

Password authentication is regarded as one of the simplest and most convenient authentication mechanisms. Conventional static password authentication methods can not resist direct wiretapping attacks, and thus, are unsuitable for open network environments. To meet today's security requirements, many password authentication methods using dynamic, or one-time, passwords have been proposed. Existing one-time password authentication schemes can be categorized into two types [1][4][5]-[8] [15][16], one may use weak passwords and the other requires strong passwords. A strong password is a password with high entropy, and thus can not be guessed easily. On the other hand, a weak password is a password with low entropy, and is easily guessable. In practice, Microsoft TechNet specifies a set of complexity criteria for a strong password [11]: (1) at least seven characters long; (2) does not contain user name, real name, or company name; (3) does not contain a complete dictionary word; (4) significantly different from previous passwords; (5) must contain characters from uppercase letters, lowercase letters, numerals, and symbols found on the keyboard. However, these criteria are only necessary conditions for choosing a strong password. Actually, a password satisfying these criteria may still be considered as a weak password, e.g., '*Hello2U!*'. Many researches, e.g., IEEE P1363.2 [3], indicate that public key cryptography is fundamental for designing secure weak-password authentication schemes. In contrast, most existing strong-password authentication schemes employ only simple operations, e.g., cryptographic hash function [13] and XOR (exclusive-or) operation. In general, the strong-password authentication schemes have the advantages over the weak-password authentication schemes in that their computational overhead are lighter, designs are simpler, and implementations are easier, and therefore are especially suitable for some constrained environments. Inevitably, using strong password increases the memory burden of the user. However, a password that is difficult to guess by the adversary is not necessarily difficult to memorize by its owner.

The first well-known strong-password authentication scheme was proposed by Lamport [8]. This scheme allows the server to authenticate the user in a way that neither eavesdropping on an authentication exchange nor reading server's database enables someone to impersonate the user. However, high hash overhead and the necessity for password resetting decrease its suitability for practical use. Additionally, Lamport's scheme is vulnerable to the replay attack. Later, Haller [5] proposed a deployed version of Lamport's scheme, the S/KEY. Like Lamport's scheme, S/KEY is also vulnerable to the replay attack [12]. To eliminate the drawbacks of Lamport's scheme and S/KEY, Shimizu

[15] proposed a one-time password authentication scheme, CINON. The one-time characteristic is gained by using two variable random numbers that are changed at each authentication. However, the user has to either memorize two variable random numbers or carry with some sort of portable storage tokens, e.g., floppy disks or IC cards. This inconvenience obstructs the deployment of CINON. Next, Shimizu et al. [16] proposed a token-free one-time password authentication scheme, PERM. The user doesn't need to either memorize any random number or carry with a portable storage token. Instead, a random number is stored in the server for authenticating the user. It is only when the server receives the correct reply corresponding to the sent random number, he will believe that the user is authentic and then refresh the stored random number. Unfortunately, PERM is subject to the man-in-the-middle attack in that the adversary can impersonate user by modifying two consecutive sessions between the user and the server.

In 2000, Sandirigama, Shimizu, and Noda [14] proposed a simple strong-password authentication scheme, SAS, which was claimed to be superior to several well-known similar schemes, e.g., S/KEY, CINON, and PERM, in storage utilization, processing time, and transmission overhead. However, SAS was found to be vulnerable to a replay attack and a denial-of-service attack [9]. Then, Lin, Sun, and Hwang [9] proposed a refined scheme, OSPA, which was asserted to be secure against the stolen-verifier attack, the replay attack, and the denial-of-service attack. Unfortunately, Chen and Ku [2] showed that OSPA and SAS can not effectively withstand a stolen-verifier attack. Furthermore, Tsuji and Shimizu [17] showed that OSPA suffers from an easier attack, the man-in-the-middle attack. Recently, Lin, Shen, and Hwang [10] proposed an improved version of OSPA, denoted by Lin-Shen-Hwang's scheme for short, and showed that it can resist the guessing attack, the replay attack, the impersonation attack, and the stolen-verifier attack. However, we find that Lin-Shen-Hwang's scheme is still vulnerable to a denial-of-service attack and a replay attack. In this paper, we will show the ways to mount these two simple attacks on Lin-Shen-Hwang's scheme.

2. Review of Lin-Shen-Hwang's Scheme

For reader's convenience, we briefly describe Lin-Shen-Hwang's scheme before demonstrating its weaknesses. The notations used in Lin-Shen-Hwang's scheme are summarized in Table 1.

Table 1. Notations of Lin-Shen-Hwang's scheme

Notation	Description
A	the user
S	the server
P	user's password
N	a random nonce
h	a cryptographic hash function
$\oplus$	bitwise XOR operation
$\|\|$	concatenation operation
x	server's secret key
E	the adversary
$U_1 \Rightarrow U_2$: *mesg*	U_1 sends *mesg* to U_2 through a secure channel
$U_1 \rightarrow U_2$: *mesg*	U_1 sends *mesg* to U_2 through a common channel

Lin-Shen-Hwang's scheme involves two phases, the registration phase and the authentication phase, which can be described as in the following.

Registration Phase

The registration phase is invoked only once for registering each user.

Step R1. $A \Rightarrow S$: $A, h^2(P \oplus N)$

Step R2. $S \Rightarrow A$: $K\ (= h^2(P \oplus N) \oplus h(x \,||\, A)), N$

In Step R1, the user A calculates $h^2(P \oplus N)$ and sends it along with his identity to the server S through a secure channel. Then, S stores the verifier $h^2(P \oplus N)$ in his database. In Step R2, S issues a smart card storing $K\ (= h^2(P \oplus N) \oplus h(x \,||\, A))$ and N to A through a secure channel.

Authentication Phase

The authentication phase is invoked whenever the user logins the authentication server.

Step A1. A uses his smart card to compute c_1, c_2, and c_3.

Step A2. $A \rightarrow S$: A, c_2, c_3

In Step A1, A inserts his smart card into a login device and keys in his password P, and then the smart card performs the following computations:

$$c_1 = K \oplus h^2(P \oplus N) = h(x \,||\, A) \quad (1)$$

$$c_2 = c_1 \oplus h(P \oplus N) \quad (2)$$

$$c_3 = h(c_1) \oplus h^2(P \oplus N') \quad (3)$$

where N' is a random nonce newly generated by A. Next, A sends $\{A, c_2, c_3\}$ to S in Step A2. After receiving A's login request, S computes $h(x \,||\, A)$, and then uses the computed $h(x \,||\, A)$ and the received c_2 to compute

$$v = h(x \,||\, A) \oplus c_2 = h(P \oplus N) \quad (4)$$

If $h(v)$ equals the stored verifier $h^2(P \oplus N)$, S grants A's login request and computes

$$h^2(P \oplus N') = h^2(x \,||\, A) \oplus c_3 \quad (5)$$

Then, S updates the verifier $h^2(P \oplus N)$ with $h^2(P \oplus N')$ for A's next login.

In the following two sections, we will show that Lin-Shen-Hwang's scheme is vulnerable to a denial-of-service attack and a replay attack.

3. Denial-of-Service Attack on Lin-Shen-Hwang's Scheme

A denial-of-service attack is an offensive action whereby the adversary could use some method to work upon the server so that the access requests issued by the legitimate user will be denied by the server. During Step A2 of Lin-Shen-Hwang's scheme, E can replace the transmitting c_3 with an equal-sized random number, denoted by R while the transmitting A and c_2 are left unchanged. After receiving this modified message, S will compute

$$v = h(x \,||\, A) \oplus c_2 = h(P \oplus N) \quad (6)$$

where $c_2 = h(x \,||\, A) \oplus h(P \oplus N)$. Since $h(v)$ equals the stored verifier $h^2(P \oplus N)$, S will grant A's login request and compute $h^2(x \,||\, A) \oplus R$.

Then, S updates the verifier $h^2(P \oplus N)$ with $h^2(x \,||\, A) \oplus R$ for A's next login. Although A can successfully login S in this session, his succeeding login requests will be denied unless he re-registers to S again. That is, E can easily lock the account of any user without using any cryptographic technique. Thus, Lin-Shen-Hwang's scheme is vulnerable to a denial-of-service attack.

4. Replay Attack on Lin-Shen-Hwang's Scheme

Suppose that, before A's nth login, the adversary E has eavesdropped A's two previous authentication messages $(A, c_2^{(n-2)}, c_3^{(n-2)})$ and $(A, c_2^{(n-1)}, c_3^{(n-1)})$. During A's nth login process, E can replace the transmitting $(A, c_2^{(n)}, c_3^{(n)})$ with $(A, c_2^{(n)}, c_3^{(n-2)})$, i.e., $c_3^{(n)}$ is replaced with $c_3^{(n-2)}$ $(= h^2(x \,||\, A) \oplus h^2(P \oplus N^{(n-1)}))$, which was used in A's $(n\text{-}2)$th authentication session. Clearly, we have $N^{(n-2)\prime} = N^{(n-1)}$. Next, S will compute

$$v^{(n)} = h(x \,||\, A) \oplus c_2^{(n)} = h(P \oplus N^{(n)}) \quad (7)$$

where $c_2^{(n)} = h(x \,||\, A) \oplus h(P \oplus N^{(n)})$. Since $h(v^{(n)})$ equals the stored verifier $h^2(P \oplus N^{(n)})$, S will grant A's login request and compute

$$\begin{aligned} & h^2(x \,||\, A) \oplus c_3^{(n-2)} \\ &= h^2(x \,||\, A) \oplus h^2(x \,||\, A) \oplus h^2(P \oplus N^{(n-1)}) \\ &= h^2(P \oplus N^{(n-1)}) \end{aligned} \quad (8)$$

Next, S replaces the verifier $h^2(P \oplus N^{(n)})$ with $h^2(P \oplus N^{(n-1)})$ for A's next login. Before A's next login, E can impersonate as A to login S by sending $(A, c_2^{(n-1)}, c_3^{(n-1)})$ to S. Because $h(h(x \,||\, A) \oplus c_2^{(n-1)})$ equals the stored verifier $h^2(P \oplus N^{(n-1)})$, S will grant E's login request and replace the verifier $h^2(P \oplus N^{(n-1)})$ with $h^2(P \oplus N^{(n)})$. In addition, E can impersonate as A to login S by using $(A, c_2^{(n)}, c_3^{(n-2)})$ as his next authentication message. Similarly, E can repeatedly use $(A, c_2^{(n-1)}, c_3^{(n-1)})$ and $(A, c_2^{(n)}, c_3^{(n-2)})$ to impersonate as A to login S in his succeeding login requests. Once E has obtained the resources or services he needs, he can send $(A, c_2^{(n)}, c_3^{(n)})$ instead of $(A, c_2^{(n)}, c_3^{(n-2)})$ to S. If this replay attack is completed before A's next login, it will not be detected easily by A.

Note that the above attack scenario is merely an instance of the replay attack that can be mounted on Lin-Shen-Hwang's scheme, and its variants can be generalized as in [17].

5. Conclusion

To achieve better efficiency, many password authentication schemes employ hash functions as their basic building blocks. So far, many strong-password authentication schemes have been proposed. Unfortunately, most of these schemes have been found insecure. Herein, we have shown that a new strong-password authentication scheme, Lin-Shen-Hwang's scheme, is vulnerable to a denial-of-service attack and a replay attack. In particular, these two simple attacks can be easily performed without compromising the server in advance.

Acknowledgment

This research was supported by the National Science Council, Republic of China, under Grant NSC-92-2213-E-030-013.

References

[1] S. Bellovin and M. Merritt, "Augmented encrypted key exchange: A password-based protocol secure against dictionary attacks and password-file compromise," in *ACM Conference on Computer and Communications security*, pp. 244–250, 1993.

[2] C. M. Chen and W. C. Ku, "Stolen-verifier attack on two new strong-password authentication protocols," *IEICE Transactions on Communications*, vol. E58-B, no. 11, pp. 2519–2521, Nov. 2002.

[3] IEEE P1363.2 / D10 (Standard specifications for public key cryptographic: password-based techniques), *IEEE P1363 working group*, July 2003.

[4] L. Gong, "Optimal authentication protocols resistant to password guessing attacks," *Proc. 8th IEEE Computer Security Foundation Workshop*, pp. 24–29, 1995.

[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] D. Jablon, "Strong password-only authenticated key exchange," *ACM Computer Communications Review*, vol. 20, no. 5 pp. 5–26, 1996.

[7] S. Keung and K. Y. Siu, "Efficient protocols secure against guessing and replay attacks," in *Proceedings of the 4th International Conference on Computer Communications and Networks*, pp. 105–112, 1995.

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

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

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

[11] Microsoft TechNet: Products & Technologies/Windows Server 2003/ Product Documentation/Security/Authentication/Passwords/ Concepts (http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/windowsserver2003/proddocs/server/windows_password_tips.asp).

[12] 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.

[13] National Institute of Standards and Technology, "Secure hash standard," *FIPS Publication 180-1*, April 1995.

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

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

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

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