A Simple Attack on a Recently Introduced Hash-Based Strong-Password Authentication Scheme

Minho Kim and Çetin Kaya Koç
(Corresponding author: Minho Kim)

School of EECS, Oregon State University
Corvallis, Oregon 97331, USA (Email: {mhkim, koc}@ece.orst.edu)

(Received June 11, 2005; revised and accepted June 27, 2005)

Abstract

The user authentication is an important part of network security. Several strong-password authentication protocols have been introduced, but a secure scheme, which probably withstands to several known attacks, is not yet available. Recently, a hash-based strong-password authentication scheme was described in [2], which withstands to the several attacks, including replay, password-file compromise, denial-of-service, and insider attacks. However, we show that this protocol is still vulnerable to stolen-verifier, denial-of-service, replay, and impersonation attacks.

Keywords: Hash function, password authentication, stolen-verifier attack, denial-of-service attack, replay attack, impersonation attack.

1 Introduction

Password-based authentication mechanisms are the simplest and most convenient way to have a user authenticated in order to provide services of a computing or communication system to a pre-selected group of authorized users. These mechanisms are less costly than the biometric methods of authentication, such as fingerprint, iris scan, voice signature, etc. A generic password-based authentication system usually hashes the password of the user with the help of hash function derived from a secret-key cryptographic function, such as MISTY, DES, or FEAL [7, 8, 12]. The hashed password is stored on the server in order to preclude stealing the password by the adversary.

Unfortunately, there are two limitations in password-based authentication systems: 1) the user must submit the bare password at every authentication, and 2) the transmitted password could be stolen by wiretapping or sniffing. One of the remedying is found the use of one-time password method by Lamport [4], but there are some practical difficulties in implementing this method, such as the problems of high overhead and password resetting. Another related method is CINON [10] which solves these problems, but it requires two random numbers generated by the user, which must be stored by the user in some sort of mobile memory device. On the other hand, the PERM (Privacy Enhanced Information Reading and Writing Management) Protocol [11] stores one random number at the host, which is sent to the user for authentication. However, there are some security flaws in such a system; the adversary can launch a man-in-the-middle attack if he can obtain the logs of two consecutive sessions.

The SAS protocol proposed in [9] is a simple strong-password authentication scheme, which is superior to several well-known schemes. But, it was shown in [5] that the SAS protocol is vulnerable to the replay attack and the denial of service attack. The OSPA (Optimal Strong-Password Authentication) Protocol given in [5] was claimed to be secure against stolen-verifier attacks, replay attacks, and the denial of service attacks. Nevertheless, it was shown in [1] the SAS and OSPA protocols cannot resist to the stolen-verifier attack as claimed. Also, an impersonation attack was described in [13] on the OSPA method without an active attack on the server. Later on, an enhanced OSPA protocol was introduced in [6], which resists to the guessing, reply, impersonation, and stolen-verifier attacks. However, it was shown in [3] that the protocol is still vulnerable to reply and denial-of-service attacks. Furthermore, these two simple attacks can easily be launched without compromising the server in advance.

Recently, a hash-based strong-password authentication scheme was described in [2], which withstands to the several attacks, including replay, password-file compromise, denial-of-service, and insider attacks. The purpose of this paper is to show that, unfortunately, this scheme is

vulnerable to stolen-verifier, impersonation, replay, and denial-of-service attacks.

First, we give the basic definitions of these attacks and describe the hash-based strong-password authentication scheme introduced in [2], and finally explain the details of our attacks.

Stolen-verifier attack. The server stores verifiers of users' passwords instead of the clear text of passwords. In the stolen-verifier attack, the adversary who has stolen the password-verifier from the server uses it directly to masquerade as a legitimate user.

Impersonation attack. This attack deceives the identity of one of the legitimate parties. An attacker inserts a message and claims that it comes from a real sender.

Replay attack. A replay attack is an offensive action in which the adversary impersonates or deceives another legitimate participant through the reuse of information obtained in a protocol. It indicates an attempt by an unauthorized third party to record the exchanged messages.

Denial of service attack. This attack is characterized by an explicit attempt which an attacker made to prevent legitimate users of a service from using that service. These attempts consist of several different flavors: disrupting service to a specific system or user, preventing a particular user accessing a service, or denying requests issued by a legitimate user.

2 A Hash-Based Strong-Password Authentication Scheme

The hash-based strong-password authentication scheme described in [2] comes with two protocols: the registration protocol and the login protocol. We introduce the notation used to describe the protocols below and explain the detailed steps of both of these protocols.

2.1 Notations

- U denotes the User, S denotes the Server, and A denotes the Adversary.
- h denotes a cryptographic hash function. $h(m)$ means the message m is hashed once, while $h^2(m)$ means m is hashed twice, that is $h^2(m) = h(h(m))$.
- N denotes an integer starting from 1 since U's initial registration.
- P denotes the strong password of U.
- K_S denotes the secret-key of S.
- T denotes the most recent time U initially registered or re-registered at S.
- $\oplus$ denotes the bitwise XOR operation, and $||$ denotes the concatenation.
- The expression $A \longrightarrow B: X$ means A sends the message X to B via an insecure channel.
- The expression $A \Longrightarrow B: X$ means A sends the message X to B via a secure channel.

2.2 Registration Protocol

This protocol is invoked whenever U initially registers or re-registers to S.

R1. U sends his registration request to S.

R2. $S \longrightarrow U: N, T$.
S sets T as the currently value of the time. If this is U's initial registration, S sets $N = 1$, otherwise S sets $N = N + 1$. Next, S sends N and T to U.

R3. $U \Longrightarrow S: h^2(S||P||N||T)$.
U computes the verifier $h^2(S||P||N||T)$ and sends it to S.

R4. S computes the user storage key $K_U^{(T)} = h(U||h(K_S||T))$ and the sealed verifier $sv^{(N)} = h^2(S||P||N||T) \oplus K_U^{(T)}$, and then he stores $sv^{(N)}$, N, and T in the password file.

2.3 Login Protocol

This protocol is invoked whenever U logins to S.

L1. U sends his login request to S.

L2. $S \longrightarrow U: r, n, t$.
S selects a random nonce r and retrieves the values of $n = N$ and $t = T$ from $S's$ password file.

L3. $U \longrightarrow S: c_1, c_2, c_3$.
U sends c_1, c_2, and c_3 to S, where

$$\begin{aligned} c_1 &= h^2(S||P||n||t) \oplus h(S||P||n||t), \\ c_2 &= h(S||P||n||t) \oplus h^2(S||P||n+1||t), \\ c_3 &= h(h^2(S||P||n+1||t)||r). \end{aligned}$$

L4. S computes $K_U^{(t)} = h(U||h(K_S||t))$, and then derives $h^2(S||P||n||t)$ from the stored sealed verifier $sv^{(n)}$ using

$$h^2(S||P||n||t) = sv^{(n)} \oplus K_U^{(t)}.$$

Then, S computes u_1 and u_2 using

$$\begin{aligned} u_1 &= c_1 \oplus h^2(S||P||n||t) = h(S||P||n||t), \\ u_2 &= c_2 \oplus u_1 = h^2(S||P||n+1||t). \end{aligned}$$

If the equalities $h(u_1) = h^2(S||P||n||t)$ and $h(u_2||r) = c_3$ hold, then S authenticates U. Otherwise, S rejects U's login request and terminates the session.

After a successful authentication, S computes a new sealed-verifier using

$$sv^{(n+1)} = u_2 \oplus K_U^{(t)} = h^2(S||P||n+1||t) \oplus K_U^{(t)},$$

and replaces $sv^{(n)}$ with $sv^{(n+1)}$, and sets $N = n+1$ for U's next login protocol. The value of T is unchanged, i.e., $T = t$.

3 Our Attack

We devise an attack assumption that the adversary steals a copy of user's password-verifier $h^2(S||P||N||T)$. Such scenarios are considered in other paper [1].

The second assumption we make is that A is capable of blocking the communication from U to S. After having stolen a copy of the password verifier, A launches an attack whenever it can block communication.

Therefore, our attack assumes that a stolen-verifier attack (by obtaining a copy of the password verifier) and a denial-of-service attack (by blocking the communication from U to S) have succeeded. We then show that under these two assumptions (attacks), the attacker can now successfully login to the system using replay, impersonate the user, and thus succeed in the impersonation attack.

Below we describe our attack step by step.

1) A steals a copy of U's password-verifier $h^2(S||P||N||T)$.

2) During the U's nth login process, A monitors the communication channel, and then he sees the request U made to S and the values r, n, and t sent by S. Next, A captures the values of c_1, c_2, and c_3 sent by U to S and blocks the communication channel from U to S. These values are not reaching to S by blocking communication.

3) A computes $h(S||P||n||t)$ and $h^2(S||P||n+1||t)$ with the help of the captured values c_1, c_2, and the previously stolen password-verifier $h^2(S||P||N||T)$ as

$$\begin{aligned} h(S||P||n||t) &= c_1 \oplus h^2(S||P||n||t), \\ h^2(S||P||n+1||t) &= c_2 \oplus h(S||P||n||t), \end{aligned}$$

where $N = n$ and $T = t$.

4) Next, A sends c_1, c_2, and c_3 to S.

5) After receiving this message, S retrieves t from the password file and computes

$$K_U^{(t)} = h(U||h(K_S||t))$$

and then uses $K_U^{(t)}$ to compute the verifier $h^2(S||P||n||t)$ with the help of the stored sealed verifier $sv^{(n)}$ as

$$h^2(S||P||n||t) = sv^{(n)} \oplus K_U^{(t)}.$$

6) Next, S computes

$$\begin{aligned} u_1 &= c_1 \oplus h^2(S||P||n||t) = h(S||P||n||t), \\ u_2 &= c_2 \oplus u_1 = h^2(S||P||n+1||t). \end{aligned}$$

If $h(u_1) = h^2(S||P||n||t)$ and $h(u_2||r) = c_3$ hold, S is supposed to authenticate the sender. Since these equalities will hold, S authenticates A as being U. Therefore, S allows the attacker A to login.

7) After this successful login, S updates the sealed verifier according to the step L4 of login protocol. Therefore, the following will be executed by S. S computes

$$sv^{(n+1)} = u_2 \oplus K_U^{(t)} = h^2(S||P||n+1||t) \oplus K_U^{(t)},$$

and replaces $sv^{(n)}$ with $sv^{(n+1)}$, and then he sets $N = n+1$ for U's next login protocol. The value of T is unchanged, i.e., $T = t$.

At the end of step 6, the adversary has successfully logged into the system impersonating the legitimate user. It can now launch other attacks within the system or access to sensitive documents. If the user logs in after the attacker does, it may not be possible to discover that the attacker has logged into the system impersonating the user, unless the user checks the login records. Until the time when the user or the system managers discover the attacker's successful login, the attacker can continue to impersonate the user.

4 Conclusions

In this paper, we have shown that a hash-based strong-password authentication scheme proposed in [2] is vulnerable if the attacker is able to obtain a copy of the verifier (stolen-verifier attack) and briefly block the communication from the user to the server (denial-of-service attack). Until the legitimate user or the system manager is able to notice the attack, the attacker can impersonate the user. If the time between two consecutive logins takes long, then the attacker is expected to inflict considerable damage by violating the security principles.

References

[1] C. M. Chen and W. C. Ku, "Stolen-verifier attack on two new strong-password authentication protocols," *IEICE Transactions on Communications*, vol. E85-B, no. 11, pp. 2519-2521, Nov 2002.

[2] W. C. Ku, "A hash-based strong-password authentication scheme without using smart cards," *ACM Operating System Review*, vol. 38, no. 1, pp. 29-34, Jan 2004.

[3] W. C. Ku, H. C. Tsai, and S. M. Chen, "Two simple attacks on Lin-Shen-Hwang's strong-password authentication protocol," *ACM Operating System Review*, vol. 37, no. 4, pp. 26-31, Oct 2003.

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

[5] 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, Sep 2001.

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

[7] M. Matsui, "New block encyrption algorithm MISTY," *Fast Software Encryption*, Springer Verlag, LNCS Nr. 1267, pp. 54-68, 1997.

[8] National Institute for Standards and Technology, "Data Encryption Standard (DES)," *FIPS 46-3*, Oct 1999.

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

[10] A. Shimizu, "A dynamic password authentication method by one-way function," *IEICE Transactions*, vol. E73-DI, no. 7, pp. 630-636, Jul 1990.

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

[12] A. Shimizu and S. Miyaguchi, "Fast data encipherment algorithm FEAL," *IEICE Transactions*, vol. J70-D, no. 7, pp. 1413-1423, Jul 1987.

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

Minho Kim is a Ph.D. student in the Department of Electrical Engineering and Computer Science at Oregon State University. He received B.S. degree in Computer Science from Korea Air Force Academy and M.S. degree in Computer Science from Yonsei University, Seoul, South Korea, in 1993 and 1998, respectively. He has also worked as an assistant professor of Computer Science at Korea Air Force Academy. His research interests are in cryptography, computer and network security, and wireless communications.

Çetin Kaya Koç is currently a professor of Electrical Engineering and Computer Science at Oregon State University. He received his Ph.D. degree from University of California, Santa Barbara. Dr. Koç's research interests are in cryptographic engineering, algorithms and architectures for cryptography, computer arithmetic and finite fields, parallel algebraic computation, and network security. He has founded the Workshop on Cryptographic Hardware and Embedded Systems (CHES), and has been an Associate Editor of IEEE Transactions on Computers and IEEE Transactions on Mobile Computing. Dr. Koç has also been working as a consulting engineer with research and development interests in cryptographic engineering and embedded systems for several companies including Intel, RSA Security, and Samsung Electronics.