Cryptanalysis of a New Password Authentication Scheme Using Hash Functions

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

Min-Hung Chiang
Department of Computer Science and Information Engineering
Fu Jen Catholic University
Email: grievous@wcku1.csie.fju.edu.tw

Shen-Tien Chang
Department of Computer Science and Information Engineering
Fu Jen Catholic University
Email: figo@wcku1.csie.fju.edu.tw

Abstract*-In 2000, Peyravian and Zunic proposed an efficient hash-based password authentication scheme that can be easily implemented. Later, Lee, Li, and Hwang demonstrated that Peyravian-Zunic's scheme is vulnerable to an off-line guessing attack, and then proposed an improved version. However, Ku, Chen, and Lee pointed out that their scheme can not resist an off-line guessing attack, a denial-of-service attack, and a stolen-verifier attack. Recently, Yoon, Ryu, and Yoo proposed an improved scheme of Lee-Li-Hwang's scheme. Unfortunately, we find that Yoon-Ryu-Yoo's scheme is still vulnerable to an off-line guessing attack and a stolen-verifier attack. Furthermore, their scheme can not achieve backward secrecy. Herein, we first briefly review Yoon-Ryu-Yoo's scheme and then describe its weaknesses.*

Keywords: Password authentication, hash function, guessing attack, denial-of-service attack, stolen-verifier 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 dynamic one-time password authentication schemes can be categorized into two types, one employs cryptosystems, either public-key cryptosystems or secret-key cryptosystems, and the other employs only one-way hash functions, e.g., [11][13], and the XOR (exclusive-or) operation. Although the latter type, the hash-based password authentication scheme, e.g., [3][8][10][14][15][16], usually requires that users should choose strong passwords, which can not be easily guessed, it has the advantage over the former type, e.g., [2], in that its computation is lighter, design is simpler, and implementation is easier, and therefore is especially suitable for certain constrained environments.

The first well-known hash-based 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 a replay attack. Later, Haller [3] proposed a deployed version of Lamport's scheme, the S/KEY. Like Lamport's scheme, S/KEY is also vulnerable to a replay attack. 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 a man-in-the-middle attack in that the adversary can impersonate user by modifying two consecutive sessions between the user and the server. However, by using weak passwords, none of the above mentioned hash-based password authentication schemes can resist the password guessing attack.

In 2000, Sandirigama et al. [14] proposed a simple hash-based strong-password authentication scheme, SAS, which was intended 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 has been found to be vulnerable to a replay attack and a denial-of-service attack [5][10]. In addition, Lin et al. [10] also proposed a refined scheme, the OSPA (Optimal Strong-Password authentication) scheme, which was asserted to be secure against the stolen-verifier attack, the replay attack, and the denial-of-service attack. Unfortunately, the OSPA scheme is also found to be vulnerable to a stolen-verifier attack [1] and a man-in-the-middle attack [17]. Independently, Peyravian and Zunic [12] also proposed a hash-based password authentication scheme. Since the associated operations are relatively simple, their scheme is efficient and can be easily implemented. Later, Hwang and Yeh [4] showed that Peyravian-Zunic's scheme is vulnerable to an off-line guessing attack, a server spoofing attack, and a stolen-verifier attack, and then proposed a modified version, which additionally uses the public-key cryptosystem. Clearly, Hwang-Yeh's scheme violates the original expectation that only simple operations are used. Moreover, it has been found [6] that Hwang-Yeh's scheme has several weaknesses.

In 2002, Lee, Li, and Hwang [9] proposed another improvement of the Peyravian-Zunic's scheme, and claimed that their scheme is secure against the off-line guessing attack. However, Ku, Chen, and Lee [7] pointed out that Lee-Li-Hwang's scheme is still vulnerable to an off-line guessing attack, a denial-of-service attack, and a stolen-verifier attack [1][10]. Recently, Yoon, Ryu, and Yoo [18] also demonstrated that Lee-Li-Hwang's scheme suffers from a denial-of-service attack, and then proposed an improved scheme of Lee-Li-Hwang's scheme by adding integrity protection to the authentication messages. Yoon-Ryu-Yoo's scheme was claimed to be resistant to the guessing attack, the replay attack, the server-spoofing attack, and the denial-of-service attack. Unfortunately, we find that Yoon-Ryu-Yoo's scheme is still vulnerable to an off-line guessing attack and a stolen-verifier attack. Furthermore, their scheme fails to achieve backward secrecy. In this article, we will describe the weaknesses of Yoon-Ryu-Yoo's scheme.

2. Review of Yoon-Ryu-Yoo's Scheme

In 2004, Yoon, Ryu, and Yoo [18] proposed a hash-based password authentication scheme, Yoon-Ryu-Yoo's scheme, which was claimed to be an improved version of Lee-Li-Hwang's scheme. Before demonstrating the weaknesses of Yoon-Ryu-Yoo's scheme, we first briefly review it for the reader's convenience. The notations used throughout this paper can be summarized as follows:

- C denotes the client.
- S denotes the server.
- E denotes the adversary.
- id denotes the identity of C.
- pw denotes the password of C.
- r_c represents the random number generated by C.
- r_s represents the random number generated by S.
- H represents a hash function.
- $\oplus$ represents the bitwise XOR operation.

Yoon-Ryu-Yoo's scheme involves three phases, the registration phase, the user authentication phase, and the change password phase, which can be described as in the following.

Registration Phase

This registration phase is invoked when C requests to register with S.

Step 1. C chooses his password hw to compute $hpw = H(id, pw)$, and then submits id and hpw to S through a secure channel. Next, S stores id and hpw, which is used as the verifier for pw, in his verification table.

User Authentication Phase

The user authentication phase is invoked whenever C requests to access the resources at S by using pw, which can be used to compute hpw.

Step 1. $C \rightarrow S$: id, $r_c \oplus hpw$, $H(r_c)$

Step 2. $C \leftarrow S$: $r_s \oplus hpw$, $H(r_c, r_s)$

Step 3. $C \rightarrow S$: id, $H(hpw, r_c, r_s)$

Step 4. $C \leftarrow S$: *access granted/denied*

If the $H(hpw, r_c, r_s)$ received in Step 3 equals the expected one, S accepts C's request and sends 'access granted' to C in Step 4. Otherwise, S rejects C's request and sends 'access denied' to C in Step 4.

Change Password Phase

The change password phase is invoked whenever C requests to change pw, which can be used to compute hpw, with a new one, say pw_{new}. The steps are the same as within the user authentication phase except that Step 3 is replaced by Step 3′ as in the following:

Step 3′. $C \rightarrow S$: id, $H(hpw, r_c, r_s)$, $Mask$, V_Mask

where $Mask = hpw_{new} \oplus H(hpw, r_c+1, r_s)$, $V\_Mask = H(hpw_{new}, r_s)$, and $hpw_{new} = H(id, pw_{new})$. If the received $H(hpw, r_c, r_s)$ in Step 3′ does not equal the excepted one, S rejects C's request and sends 'access denied' to C in Step 4. Otherwise, S computes $H(hpw, r_c+1, r_s)$ and then uses the result to retrieve hpw_{new} from the received $Mask$ (= $hpw_{new} \oplus H(hpw, r_c+1, r_s)$). Next, S computes $H(hpw_{new}, r_s)$ and if the result equals V_Mask, S replaces the verifier hpw with hpw_{new} and sends 'access granted' to C in Step 4. Otherwise, S rejects C's request and sends 'access denied' to C in Step 4.

3. Weaknesses of Yoon-Ryu-Yoo's Scheme

In this section, we will demonstrate the ways to mount an off-line guessing attack and a stolen-verifier attack [1,6] on Yoon-Ryu-Yoo's scheme. Furthermore, we will also show that Yoon-Ryu-Yoo's scheme can not achieve backward secrecy.

Off-Line Guessing Attack

Suppose that the adversary E has intercepted id, $r_c \oplus hpw$, and $H(r_c)$ in a previous run of the user authentication phase. E can guess a password pw' and then compute

$$hpw' = H(id, pw').$$

Next, he can compute

$$H(r_c)' = H(r_c')$$

where

$$r_c' = (r_c \oplus hpw) \oplus hpw'.$$

If the computed $H(r_c)'$ equals the intercepted $H(r_c)$, E has correctly guessed C's password, i.e., $pw' = pw$. Otherwise, E tries another guess for pw. Thus, Yoon-Ryu-Yoo's scheme fails to effectively resist the off-line guessing attack as its authors claimed.

Stolen-Verifier Attack

Suppose that the adversary E has stolen the verifier hpw (= $H(id, pw)$). Clearly, since E can derive pw by employing an off-line guessing attack, in which each guess for pw can be verified by using the stolen hpw, Yoon-Ryu-Yoo's scheme is vulnerable to a stolen-verifier attack [1,6]. Alternatively, Yoon-Ryu-Yoo's scheme suffers from another form of the stolen-verifier attack as follows. In the user authentication phase, E can randomly select r_E to compute

$$r_E \oplus hpw$$

$$H(r_E)$$

and send $\{id, r_E \oplus hpw, H(r_E)\}$ to S in Step 1. Next, S retrieves r_E from the second item of the received message by using hpw, and then computes $H(r_E)$. As the computed $H(r_E)$ equals the third item of the received message, r_E is verified. Then, S randomly selects r_s to compute

$$r_s \oplus hpw$$

$$H(r_E, r_s)$$

and sends $\{r_s \oplus hpw, H(r_E, r_s)\}$ to E in Step 2. Next, E can retrieve r_s from the first item of the received message by using the stolen hpw, and then compute

$$H(hpw, r_E, r_s)$$

Then, E sends $\{id, H(hpw, r_E, r_s)\}$ to S in Step 3. Since the received message equals the expected one, S will accept E's request. In addition, a stolen-verifier attack can also be mounted on the change password phase in the same way except for Step 3′ as follows. E can select a password pw_E to compute $hpw_E = H(id, pw_E)$, compute

$$hpw_E = H(id, pw_E)$$

$$Mask_E = hpw_E \oplus H(hpw, r_E+1, r_s)$$

$$V\_Mask_E = H(hpw_E, r_s)$$

and then send $\{id, H(hpw, r_E, r_s), Mask_E, V\_Mask_E\}$ to S in Step 3′. Since $H(hpw, r_E, r_s)$ equals the expected one, S will compute $H(hpw, r_E+1, r_s)$ to retrieve hpw_E from the received $Mask_E$, and then compute $H(hpw_E, r_s)$. As the computed result equals V_Mask_E, S will be fooled into changing C's verifier hpw with hpw_E.

Lack of Backward Secrecy

Suppose that the adversary E has stolen hpw. If C has detected that hpw is compromised, he can invoke the password change phase to change pw with a new one, say pw_{new}. However, by intercepting the messages transmitted in Step 1 and Step 2 of the change password phase, E can use the stolen hpw to retrieve r_c and r_s and then computes $H(hpw, r_c+1, r_s)$. In addition, by intercepting the message transmitted in Step 3′ of the change password phase, E can use the computed $H(hpw, r_c+1, r_s)$ to retrieve hpw_{new} from $Mask$ (= $hpw_{new} \oplus H(hpw, r_c+1, r_s)$). Hence, E

can still impersonate C to login S by using hpw_{new}. Note that if E has learned just an ever used hpw_{old}, he can obtain the current hpw by iteratively applying the above method. Therefore, Yoon-Ryu-Yoo's scheme can not achieve backward secrecy.

4. Conclusion

Many password authentication schemes employ the hash function as their main building block for improving efficiency and reducing implementation cost. However, most of these schemes are flawed. Herein, we have demonstrated that a new hash-based password authentication scheme proposed by Yoon, Ryu, and Yoo is vulnerable to an off-line guessing attack and a stolen-verifier attack. Furthermore, Yoon-Ryu-Yoo's scheme fails to achieve backward secrecy.

Acknowledgment

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

References

[1] 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.

[2] Draft D2002-12-20 of IEEE P1363.2 (Standard specifications for public key cryptographic: password-based techniques), *IEEE P1363 working group*, 2002.

[3] N. M. Haller, "A one-time password system," *RFC 1704*, 1994.

[4] J. J. Hwang and T. C. Yeh, "Improvement on Peyravian-Zunic's password authentication schemes," *IEICE Transactions on Communications*, vol. E85-B, no. 4, pp. 823-825, April 2002.

[5] W. C. Ku and C. M. Chen, "Cryptanalysis of a one time password authentication protocols," in *Proceedings of the 2001 National Computer Symposium*, Taiwan, pp. F046-F050, Dec. 2001.

[6] W. C. Ku, C. M. Chen, and H. L. Lee, "Cryptanalysis of a variant of Peyravian-Zunic's password authentication scheme," *IEICE Transactions on Communications*, vol. E86-B, no. 5, pp. 1682-1684, May 2003.

[7] W. C. Ku, C. M. Chen, and H. L. Lee, "Weaknesses of Lee-Li-Hwang's hash-based password authentication scheme," *ACM Operating Systems Review*, vol. 37, no. 4, pp. 19-25, Oct. 2003.

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

[9] C. C. Lee, L. H. Li, and M. S. Hwang, "A remote user authentication scheme using hash functions," *ACM Operating Systems Review*, vol. 36, no. 4, pp. 23-29, Oct. 2002.

[10] 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.

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

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

[13] R. Rivest, "The MD5 message-digest algorithm," *RFC 1321*, April 1992.

[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, July 2003.

[18] E. J. Yoon, E. K. Ryu, and K. Y. Yoo, "A secure user authentication scheme using hash functions," *ACM Operating Systems Review*, vol. 38, no. 2, pp. 62-68, April 2004.