Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 6 (3): 639-643, 2006
ISSN 1812-5654

Improvement of Fixing Problems in Lin *et al.*'s OSPA Protocol

[1]Jau-Ji Shen, [2]Ching-Ying Lin and [3]Hung-Wen Yang
[1]Department of Information Management, National Formosa University,
64 Wunhua Rd., Huwei, Yunlin County, Taiwan 632, Republic of China
[2]Graduate Institute of Networking and Communication Engineeing,
Chaoyang University of Technology, 168 Gifeng E. Rd., Wufeng,
Taichung County, Taiwan 413, Republic of China
[3]Department of Computer Science, National Chung Hsing University,
250, Kuo Kuang Road,Taichung County,
Taiwan 402, Republic of China

Abstract: Recently, Yoon *et al.* proposed an improved scheme to solve the problems of replay attack and denial of service attack in the Lin *et al.*'s OSPA scheme. Their scheme can simply update user password, provides mutual authentication between the user and remote server and has more efficient performance by reducing the number of hash operations. In this study, an improved scheme is shown to be vulnerable to the insider attack and smart card loss problem and propose an enhancement of the protocol to solve these problems.

Key words: Cryptography, authentication, password, smart card, hash function

INTRODUCTION

User authentication is the basic security mechanism for remote login systems. Among numerous methods for user authentication, the password authentication is the most convenient and widely adopted. Recently, several password authentication schemes using smart card have been proposed[1-6]. Lamport[7] proposed an one-time password authentication scheme for insecure communication, his scheme uses an one way encryption function to achieve the security of system, but this method has high hash overhead and the requirement of resetting the password verifier. Recently, Sandirigama *et al.*[8] proposed a simple and secure password authentication protocol called the SAS protocol, it only needs fewer hash overhead and does not need password resetting. Later, Lin *et al.*[9] showed that the SAS protocol is vulnerable to the replay attack and the denial of service attack and then proposed an optimal strong password authentication scheme called the OSAP scheme, which is secure against stolen verifier, replay and denial of service attacks. But, Chen and Ku[10] found out that it is vulnerable to the stolen verifier attack. Later, Lin *et al.*[11] proposed an improvement to the OSAP protocol to enhance the security flaw. Unfortunately, Ku *et al.*[12] found out that Lin *et al.*'s protocol is vulnerable to the replay attack and the denial of service attack. Later, Yoon *et al.*[13] proposed an enhancement of the protocol to prevent such problems. Their scheme can simply update user password and provides mutual authentication between the user and a remote server. However, in this study, it is pointed out that the Yoon *et al.*'s scheme is insecure and propose an improved scheme that can resist the weakness of their scheme.

REVIEW OF YOON *et al.*'s SCHEME

Their scheme is composed of three phases: registration, authentication and password change phases. Before explaining their scheme, we introduce the used notations.

Notations: The notations and abbreviations used in this study are described as follows:

- A, S, E : user, server and adversary, respectively.
- P, P′: user's password and new password.
- N, N′: random nonce and new random nonce.
- vpw, new_vpw: user's password verifier and new password verifier.
- x : server's secret key.

Corresponding Author: Ching-Ying Lin, Graduate Institute of Networking and Communication Engineeing, Chaoyang University of Technology, 168 Gifeng E. Rd., Wufeng, Taichung County, Taiwan 413, Republic of China

- $\oplus$: bitwise XOR operation.
- $\|$: concatenation operation.
- $h(\cdot)$: a strong one-way hash function.

Registration phase: When a new user wants to access the resource from server, he must register to the server over a secure channel.

Step (R1): A→S: {A, P}
A sends his identity A and password P to S.

Step (R2): S→A: {X, K, N, h(·)}
After receiving {A,P}, S selects a random nonce N and computes vpw = h (P⊕N), X = h(x‖A) and K = h(x‖A)⊕P. The vpw is a password verifier stored in the server's database and {X, K, N, h(·)} are stored in the smart card which is send to A.

Authentication phase: If A wants to login the server, he must inserts his smart card into the card reader and key in his identity A and password P, then the smart card will performs the following operations:

Step (A1): A→S: {A, C_2, C_3}
Computes C_1 = K⊕P = h(x‖A) and verify whether C_1 equals to the stored X or not. If they are equal, then the smart card selects a new random nonce N′ and computes new_vpw = (P⊕N′), C_2 = new_vpw⊕h (C_1, h(P⊕N)) and C_3 = h(C_1, h(P⊕N), new_vpw), which new_vpw is a new password verifier for the next login. Then, A sends {A, C_2, C_3} to S.

Step (A2): S→A: {C_4}
Upon receiving the message {A, C_2, C_3}, S checks the identity of A and computes C'_1 = h(x‖A), new_vpw = C_2⊕h(C'_1, vpw), C'_3 = h(C'_1, vpw, new_vpw). Then, S verifies C'_3 with C_3. If they are equal, S accepts the login request and updates vpw with new_vpw for the next login. Then, S computes C_4 = h(A, C'_1, new_vpw) and sends it to A.

Step (A3): Upon receiving the message{C_4}, smart card computes C'_4=h(A,C'_1, new_vpw) and verifies C_4 with C'_4. If they are equal, A will believes that the communication part is the legitimate server and the smart card replaces the stored N with N′. Otherwise, A disconnects this connection (Fig. 1).

Password change phase: The procedures for Yoon *et al.*'s password change phase are as follows. We assume that A wants to change his password P to P′, he also inserts his smart card into the card reader and key in his identity A and password P, then the smart card will perform the following operations:

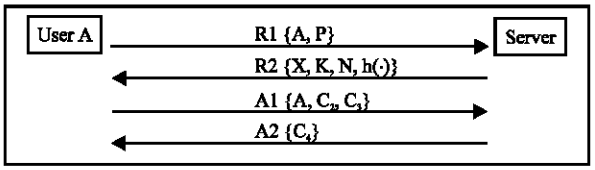

Fig. 1: The registration and authentication phase of Yoon *et al.*'s scheme

Step (C1): A→S: {A, C_2, C_3}
Computes C_1 = K⊕P = h(x‖A) and verifies whether C_1 equals to the stored X or not. If they are equal, the smart card selects a new password P′ and a new random nonce N′, then the smart card computes new_vpw = (P′⊕N′), C_2 = new_vpw⊕h(C_1, h(P⊕N)) and C_3 = h(C_1, h(P⊕N), new_vpw), which new_vpw is a new verifier for the next login. Then, A sends {A, C_2, C_3} to S.

Step (C2): S→A: {C_4}
Upon receiving the message {A, C_2, C_3}, S checks the identity of A and computes C'_1 = h(x‖A), new_vpw = C_2⊕h(C'_1, vpw), C'_3 = h(C'_1, vpw, new_vpw). Then, S verifies C'_3 with C_3. If they are equal, S accepts the login request and updates vpw with new_vpw for the next login. Finally, S computes C_4 = h(A, C'_1, new_vpw) and sends it to A.

Step (C3): Upon receiving the message {C_4}, smart card computes C'_4 = h(A,C_1, new_vpw) and verifies C_4 with C'_4. If they are equal, A will believes that the communication part is the legitimate server. Finally, the smart card replaces the stored N with N′ and stores K′ in place of K by computing K′ = C_1⊕P′ for password change.

CRYPTANALYSIS OF YOON *et al.*'s SCHEME

Yoon *et al.*'s scheme can solve the weaknesses of Lin *et al.*'s scheme, but their scheme does not prevent the insider attack and smart card loss problem. These weaknesses are presented below:

Insider attack: In practice, it is likely that A uses the same password P to access several servers for his convenience. If the insider of S has obtained P, he can impersonate A

to access other servers[14]. In the registration phase of Yoon *et al.* proposal, the user sends his/her password to the remote server with plain-text form. It is very easy to cause an insider attack because the server knows A's password and an insider attacker may get it and use it to login other servers for accessing data.

Smart card loss problem: In Yoon *et al.*'s scheme, they did not consider about the smart card loss problem. We know the smart card is stored with {X, K, N, h(·)}, if an user loses his smart card, an attacker can obtain his password P by computing X⊕K. Then, the attacker can simulate the legal user to computes {C_1, C_2, C_3} and send {A, C_2, C_3} to the server. Then, the server will believe the login request message is from the legal user and authenticate this login.

IMPROVEMENT OF YOON *et al.*'s SCHEME

To resist the insider attack and prevent the smart card loss problem, we propose an improved as follows.

Registration phase: When a new user wants to access the resource from server, he must register to the server over a secure channel.

Step (R1): A→ S: {A, vpw}

A chooses his password P and a random nonce N, then computes password verifier vpw = h(P⊕N). A sends his identity A and password verifier vpw to S.

Step (R2): S → A{C_1, h()}

Upon receiving the message {A, vpw}, S stores A's vpw in the server's database and computes X = h(x‖A), C_1 = vpw⊕X. Then, S stores {C_1, h()} into the smart card and sends it to A. After receiving the smart card, A stores the random nonce N in his smart card.

Authentication phase: When A wants to login the server, he must inserts his smart card into the card reader and key in his identity A and password P, then the smart card will performs the following operations:

Step (A1): A → S: {A, C_2, C_3}

The smart c ard retrieves X by computing C_1⊕h(P⊕N) and chooses a new random nonce N′, then the smart card computes new_vpw = h(P⊕N′), C_2 = new_vpw⊕h(X, h(P⊕N)) and C_3 = h(X, h(P⊕N), new_vpw), where the new_vpw is a new verifier for the next login. Then, A sends {A, C_2, C_3} to S.

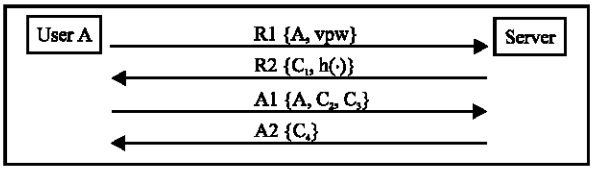

Fig. 2: The registration and authentication phase of our scheme

Step (A2): S → A: {C_4}

Upon receiving the message {A, C_2, C_3}, S checks the identity of A and computes X′ = h(x ‖ A), new_vpw = C_2⊕h(X′, vpw), C'_3 = h(X′, vpw, new_vpw). Then, S verifies C'_3 with C_3. If they are equal, S accepts the login request and updates vpw with new_vpw for the next login. Finally, S computes C_4 = h(A, X′, new_vpw) and sends it to A.

Step (A3): Upon receiving the message {C_4}, smart card computes C'_4 = h (A, X, new_vpw) and verifies C_4 with C'_4. If they are equal, A will believes that the communication part is the legitimate server. Then, the smart card replaces the stored N with N′ and updates C_1 with C'_1 by computing C_1⊕vpw⊕new_vpw. Otherwise, A disconnects this connection. The procedures are shown in the Fig. 2.

Password change phase: The procedures for our password change phases are as follows. We assume that A wants to change his password P to P′, he also inserts his smart card into the card reader and key in his identity A and password P, then the smart card will perform the following operations:

Step (C1): A → S: {A, C_2, C_3}

The smart card retrieves X by computing C_1⊕h(P⊕N) and chooses a new random password P′ and a new random nonce N′, then the smart card computes new_vpw = h(P′⊕N′), C_2 = new_vpw⊕h(X, h(P⊕N)) and C_3 = h(X, h(P⊕N), new_vpw), which new_vpw is a new verifier for the next login. A sends {A, C_2, C_3} to S.

Step (C2): S → A: {C_4}

Upon receiving the message {A, C_2, C_3}, S checks the identity of A and computes X′ = h(x ‖ A), new_vpw = C_2⊕h(X′, vpw), C'_3 = h(X′, vpw, new_vpw). Then, S verifies C'_3 with C_3. If they are equal, S accepts the login request and updates vpw with

new_vpw for the next login. Finally, S computes $C_4 = h(A, X', new\_vpw)$ and sends it to A.

Step (C3): Upon receiving the message $\{C_4\}$, smart card computes $C'_4 = h(A, X, new\_vpw)$ and verifies C_4 with C'_4. If they are equal, A will believes that the communication part is the legitimate server. Then, the smart card replaces the stored N with N′ and updates C_1 with C'_1 by computing $C_1 \oplus vpw \oplus new\_vpw$.

SECURITY ANALYSIS

Guessing attack: Suppose an adversary E intercepts a login request message $\{A, C_2, C_3\}$, E cannot derive the password P from the parameters C_2 and C_3 because E does not know the server's secret key x, random nonce N and new random nonce N′.

Replay attack: Assume the adversary E intercepts a login request message $\{A, C_2, C_3\}$, E cannot login to the remote server by replaying the message $\{A, C_2, C_3\}$ because the next password verifier is hidden in the previous session such that C_3 is an implicit verifier $new\_vpw = h(P \oplus N)$ for the next session.

Impersonal attack: If an adversary E attempts to modify a login request message $\{A, C_2, C_3\}$ into $\{A, C''_2, C''_3\}$ and sent it to the server, it can not be successful because E cannot obtain the $h(x \| A)$ and $h(P \oplus N)$ to compute the valid parameters C_2 and C_3.

Stolen-verifier attack: Assume an adversary E has stolen a password verifier $h(P \oplus N)$ from the server and intercepts the user's $(n-1)^{th}$ login request message $\{A, C_2^{n-1}, C_3^{n-1}\}$ over the public network. Because E cannot derive $h(x \| A)$ and $new\_vpw = h(P \oplus N')$ from C_2^{n-1} and C_3^{n-1} by way of the password verifier $h(P \oplus N)$.

Denial of service attack: Assume the adversary E replaces the login message $\{A, C_2, C_3\}$ to $\{A, C''_2, C''_3\}$, this attack can not be successful because a successful authentication is checked only if C'_3 is equivalent to $C_3 = h(X, h(P \oplus N), new\_vpw)$. After a successful authentication, the server updates a password verifier $h(P \oplus N)$ with new password verifier $new\_vpw = h(P \oplus N')$ for the next login.

Insider attack: In our scheme, the user sends $\{A, vpw\}$ to the server, where $vpw = h(P \oplus N)$. In such scheme, the server can not obtain the user's password that can overcome the insider attack.

Smart card loss problem: If the smart card is lost, no one can imitate the owner to login the server. Because of the information of smart card contains $\{C_1, N, h(\cdot)\}$, without any knowledge of password P.

CONCLUSION

This study showed that the Yoon *et al.*'s scheme is vulnerable to the insider attack and the smart card loss problem. Further, the proposed scheme is showed to enhance the security of Yoon *et al.*'s protocol.

ACKNOWLEDGEMENT

This research was partially supported by the National Science Council, Taiwan, R.O.C., under contract NO.NSC93-2213-E-005-033.

REFERENCES

1. Chang, C.C. and T.C. Wu, 1991. Remote password authentication with smart card. IEEE Proceedings, E138: 165-168.
2. Chang, C.C. and W.Y. Liao, 1994. A remote password authentication scheme based upon ElGamal's signature scheme. Comp. Secur., 13: 137-144.
3. Sun, H.M., 2000. An efficient remote user authentication scheme using smart cards. IEEE Trans. Consum. Electron., 46: 958-961.
4. Chien, H.Y., J.K. Jan and Y.M. Tseng, 2001. A modified remote login authentication scheme based on geometric approach. J. Sys. Software, 55: 287-290.
5. Hwang, M.S. and L.H. Li, 2000. A new remote user authentication scheme using smart cards. IEEE Trans. Consum. Electron., 46: 28-30.
6. Wu, T.C., 1995. Remote login authentication scheme based on a geometric approach. Comp. Comm., 18: 959-963.
7. Lamport, L., 1981. Password authentication with insecure communication. Comm. ACM., 24: 770-772.
8. Sandirigama, M., A. Shimizu and M.T. Noda, 2000. Simple and secure password authentication protocol (SAS). IEICE Trans. Comm., E83-B: 1363-1365.

9. Lin, C.L., H.M. Sun and T. Hwang, 2001. Attacks and solutions on strong-password authentication. IEICE Trans. Comm., E84-B: 2622-2627.
10. Chen, C.M. and W.C. Ku, 2002. Stolen-verifier attack on two new strong-password authentication protocols. IEICE Trans. Comm., E85-B: 2519-2521.
11. Lin, CW., J.J. Shen and M.S. Hwang, 2003. Security enhancement for optimal strong-password authentication protocol. ACM Operat. Sys. Rev., 37: 12-16.
12. Ku, W.C., H.C. Tsai and S.M. Chen, 2003. Two simple attacks on Lin-Shen-Hwnag's strong-password authentication protocol. ACM Operat. Sys. Rev., 37: 26-31.
13. Yoon, E.J., E.K. Ryu and K.Y. Yoo, 2005. Fixing problems in Lin *et al.*'s OSPA protocol. Applied Math. Comp., 166: 46-57.
14. Ku, W.C., C.M. Chen and H.L. Lee, 2003. Cryptanalysis of a variant of Peyravian-Zunic's password authentication scheme. IEICE Trans. Comm., E86-B: 1682-1687.