Comments on five smart card based password authentication protocols

Yalin Chen[1], Jue-Sam Chou[2,]* , Chun-Hui Huang[3]

[1] Institute of information systems and applications, National Tsing Hua University
d949702@oz.nthu.edu.tw
[2] Department of Information Management, Nanhua University, Taiwan, R.O.C
*: corresponding author
jschou@mail.nhu.edu.tw
Tel: 886+ (0)5+272-1001 ext.56536
[3] Department of Information Management, Nanhua University, Taiwan, R.O.C
g6451519@mail1.nhu.edu.tw

Abstract

In this paper, we use the ten security requirements proposed by Liao et al. for a smart card based authentication protocol to examine five recent work in this area. After analyses, we found that the protocols of Juang et al.'s, Hsiang et al.'s, Kim et al.'s, and Li et al.'s all suffer from the password guessing attack if the smart card is lost and the protocol of Xu et al.'s suffers from the insider attack.

Keywords: *password authentication protocol, insider attack, smart card lost problem, password guessing attack*

1. Introduction

Smart card based password authentication protocols [1-20] are widely adopted for logging into the remote servers. The protocols can provide mutual authentication between the client and the server over an open network. They make the users able to be authenticated by the remote server using a human-rememberable password and can provide the system with both an effective two-factor authentication mechanism and the ability of a remote server to authenticate a legal user without the necessity of maintaining a password table.

In 2006, Liao et al.[2] proposed ten security requirements for evaluating the goodness of a smart card based password authentication protocol. We show them as follows.

R1. It needs no password or verifier table.

R2. The clients can choose and change their passwords freely.

R3. The clients need not to reveal their passwords to the server.

R4. The passwords are not transmitted in plaintext over the Internet.

R5. It can resist the insider (a legal user) attack.

R6. It can resist replay attack, password guessing attack, modification-verifier-table attack, and stolen-verifier attack.

R7. The length of a password is appropriate for memorization.

R8. It is efficient and practical.

R9. It can achieve mutual authentication.

R10. It can resist password guessing attack even if the smart card is lost.

In their article, they also proposed a protocol attempting to satisfy these ten security requirements. But Xiang et al.[9] demonstrated that their protocol suffers from both the replay attack and the password guessing attack. Other than theirs, many efforts trying to propose secure protocols of this kind were made recently. For example in 2008, Juang et al.[7] proposed "Efficient password authenticated key agreement using bilinear pairings". In 2009, Hsiang et al.[14], Kim et al.[16], and Xu et al.[18] each also proposed a protocol of this kind, respectively. In this year 2010, Li et al.[20] also proposed a protocol in this area. Although they claimed their protocols are secure. However, in this paper, we will show the violations of R5 in [18] and R10 in [7], [14], [16], [20], correspondingly.

The remainder of this paper is organized as follows: In Section 2, we review and attack on the scheme of Juang et al.'s [7]. Then we review and attack on the protocols of Hsiang et al. 's [14], Kim et al.[16], Xu et al. 's [18], and Li et al. 's [20] in Section 3 through 6, respectively. Finally, a conclusion is given in Section 7.

2. Review and attack on Juang et al.'s scheme

In their protocol [7], if an attacker gets C's smart card, he can successfully launch an off-line password-guessing attack for impersonating C to log into the server S. In the following, we first review Juang *et al.*'s protocol in Section 2.1 and then show the attack in Section 2.2.

2.1 Review

Their protocol consists of four phases: the setup phase, the registration phase, the login and authentication phase, and the password changing phase.

In the setup phase, S chooses two secrets s, x and publishes $P_s=sP$, where P is a generator of an additive cyclic group G_1 with a prime order q.

In the registration phase, the server S issues to legal user i a smart card which contains b_i (b_i=E_x[H(PW_i, b), ID_i, H(H(PW_i, b), ID_i)] and E_x[M] is a ciphertext of M encrypted by S's secret key x) and b (a random number chosen by i).

When i wants to log into S, i starts the login and authentication phase and sends

{aP, α} to S, where a is a random number chosen by i, α=$E_{Ka}[b_i]$, Ka=H(aP, P_s, Q, ê(P_s, aQ)), ê: $G_1 \times G_1 \rightarrow G_2$ is a bilinear mapping, Q=h(ID_s), h(·) is a map-to-point hash function h:{0,1}*→G_1, and ID_s is S's identification. Subsequently, S chooses a random number r, computes the session key sk=H(H(aP, P_s, Q, ê(aP, sQ)), r, ID_i, ID_s) =H(Ka, r, ID_i, ID_s) since ê(P_s, aQ)= ê(aP, sQ) , and sends {$Auth_s$, r} to user i, where ID_i is i's identification, $Auth_s$=H(Ka, H(PW_i, b), r, sk), and H(PW_i, b) is obtained from decrypting α and b_i. Then, i computes the session key sk. For authenticating S, he verifies $Auth_s$ to see if it is equal to H(Ka, H(PW_i, b), r, sk). If it is, i computes and sends {$Auth_i$} to S, where $Auth_i$=H(Ka, H(PW_i, b), r+1, sk) and H(PW_i, b) is the hash result of b stored in the smart card with PW_i inputted by i. Finally, for authenticating i, S checks to see if $Auth_i$ is equal to H(Ka, H(PW_i, b), r+1, sk).

2.2 Attack

In this protocol, it can be easily seen that if user C lost his smart card and the card is got by an insider E, E can impersonate C to log into S. We show the attack in the following.

E reads out b and b_c (which equals E_x[H(PW_c, b), ID_i, H(H(PW_i, b), ID_i)]) stored in C's smart card but he doesn't have the knowledge of PW_c. He can choose a random number c, computes cP, Kc=H(cP, P_s, Q, ê(P_s, cQ)), α=$E_{Kc}[b_c]$, starts the protocol, and masquerades as C to send { cP, α} to S. After receiving the message, S chooses a random number r, computes session key sk=H(Kc, r, ID_c, ID_s), $Auth_s$=H(Kc, H(PW_c, b), r, sk), and sends {$Auth_s$, r} to C. E intercepts the message and launches an off-line password guessing attack. He chooses a possible password PW', computes Kc=H(cP, P_s, Q, ê(P_s, cQ)), sk=H(Kc, r, ID_c, ID_s), H(Kc, H(PW', b), r, sk) and checks to see if it is equal to the received $Auth_s$. If it is, the attacker successfully gets C's password PW_c which is equal to PW'. Subsequently, E can masquerade as C by using PW' and C's smart card to log into S. That is, he can successfully implement the impersonation attack and the password guessing attack if the smart card is lost.

3. Review and attack on the protocol of Hsiang et al.'s scheme

In this section, we first review Hsiang *et al*.'s protocol [14] in Section 3.1, then demonstrate the smart card loss problem in Section 3.2.

3.1 Review

In their protocol, when user C wants to change his password, he inserts his card and types his ID and PW. The smart card computes $P^*=R \oplus$H($b \oplus PW$), and V^*=H($P^* \oplus$H(PW)), and compares V^* with V, where PW is C's password inputted for being changed, and R, b, and V are stored in C's smart card. If they are equal, the card accepts the password change request and then computes $R_{new}=P^*\oplus H(b\oplus PW^*)$ and $V_{new}=H(P^*\oplus H(PW^*))$, where PW^* is a new password submitted by C. Finally, the smart card replaces V with V_{new}.

3.2 Attack

Assume that an attacker who can get C's smart card reads the values of R, b, and V and implements a password-guessing attack. He chooses a possible password PW', computes $P'=R\oplus H(b\oplus PW')$ and $V'=H(P'\oplus H(PW'))$, and checks to see if V' and V are equal. If they are, PW' is the correct password. Then, for changing the password from PW' to PW'', the attacker logins to the server and computes $R''=P'\oplus H(b\oplus PW'')$ and $V''=H(P'\oplus H(PW''))$, where PW'' is a new password submited by E. Finally, the smart card replaces R and V with R'' and V'', respectively. The attacker can therefore masquerade as C to log into the server. That is, the attacker successfully implements the impersonation attack and the password guessing attack if the smart card is lost.

4. Review and attack on the protocol of Kim et al.'s scheme

In this section, we first review Kim *et al.*'s protocol [16] in Section 4.1, then demonstrate the smart card loss problem in Section 4.2.

4.1 Review

In their protocol, when user C wants to change his password, he inserts his card and types his ID and PW. The smart card computes $K^*_1=R\oplus H(PW)$ and compares K^*_1 with K_1 to see if they are equal, where $R(=K_1\oplus H(PW_c))$ and $K_1(=H(ID\oplus x)\oplus N)$ are stored in C's smart card, PW_c is chosen by the user when he registers at the remote server S, and N is a random number. If they are, the card accepts the password change request and C inputs a new password PW^*. Then, the card computes $R^*= K^*_1\oplus H(PW^*)$ and $K^*_2= K_2\oplus H(PW\oplus H(PW))\oplus H(PW^*\oplus H(PW^*))$, where $K_2=H(ID\oplus x\oplus N)\oplus H(PW_c\oplus H(PW_c))$ is also stored in C's smart card. Finally, the smart card will replace R and K_2 with R^* and K^*_2, respectively.

4.2 Attack

An attacker who gets C's smart card and reads the values of R, K_1, and K_2 can launch a password-guessing attack. He chooses a possible password PW', computes $K'_1=R\oplus H(PW')$, and checks to see if K'_1 and K_1 are equal. If they are, PW' is the correct password. Then, for changing the password from PW' to PW^*, the attacker logins to the server and computes $R^* = K'_1 \oplus \mathrm{H}(PW^*)$ and $K_2^* = K_2 \oplus \mathrm{H}(PW' \oplus \mathrm{H}(PW')) \oplus \mathrm{H}(PW^* \oplus \mathrm{H}(PW^*))$. He then replaces R and K_2 with R^* and K_2^*, respectively. Eventually, he can masquerade as C to log into the server. That is, he can successfully implement the impersonation attack and the password guessing attack if the smart card is lost.

5. Review and attack on the protocol of Xu et al.'s scheme

We first briefly review the protocol [18] in Section 5.1 and then present our attack in Section 5.2.

5.1 Review

Xu *et al.*'s protocol consists of three phases: the registration phase, the login phase and the authentication phase.

In the registration phase, user C submits his ID_c and PW_c to the server S. S issues C a smart card which stores C's identity ID_c, and $B=\mathrm{H}(ID_c)^x+\mathrm{H}(PW_c)$, where x is S's secret key and PW_c is C's password.

In the login phase, user C inputs ID_c and PW_c to his smart card. The card obtains system's timestamp T, chooses a random number v, computes $B_c=(B\text{-}\mathrm{H}(PW_c))^v = \mathrm{H}(ID_c)^{x\,v}$, $W=\mathrm{H}(ID_c)^v$, and $C_1=\mathrm{H}(T, B_c, W, ID_c)$, and sends { ID_c, C_1, W, T } to S.

In the authentication phase, after receiving { ID_c, C_1, W, T } at time T^*, S computes $B_s= W^x$, and checks to see if ID_c is valid, $T^*-T < \triangle\mathrm{T}$, and C_1 is equal to $\mathrm{H}(T, B_s, W, ID_c)$. If they are, S selects a random number m, sets T_s to be the current time, computes $M=\mathrm{H}(ID_c)^m$, $C_s=\mathrm{H}(M, B_s, T_s, ID_c)$, and sends { ID_c, C_s, M, T_s } to C. After receiving the message, C validates ID_c and T_s, computes $\mathrm{H}(M, B_c, T_s, ID_c)$, and compares it with the received C_s. If they are equal, S is authentic. Then, C and S can compute the common session key as $sk=\mathrm{H}(ID_c, M, W, M^v)$ and $sk=\mathrm{H}(ID_c, M, W, W^m)$, respectively.

5.2 Attack

Assume that a malicious insider U wants to masquerade as C to access S's resource. He reads B from his smart card, obtains system's timestamp T_u, chooses a random number r, computes $B_u=(B\text{-}\mathrm{H}(PW_u))^r = \mathrm{H}(ID_u)^{xr}$, $W=\mathrm{H}(ID_c)^r$, $C_1=\mathrm{H}(T_u, B_u, W, ID_c)$, and sends { ID_c, C_1, W, T_u } to S.

After receiving the message, S validates ID_c and T_u, computes $B_s= W^x = \mathrm{H}(ID_c)^{r\,x}$, and checks to see if the received C_1 is equal to the computed $\mathrm{H}(T_u, B_s, W, ID_c)$. In this case, we can see that C_1 is doomed to be equal to $\mathrm{H}(T_u, B_s, W, ID_c)$. So, U (who masquerades as C) is authentic. Finally, S obtains the system's timestamp T_s and sends

{ ID_c, C_s, M, T_s } to U, where M=H(ID_c)^m and m is a random number chosen by S. U also can compute the session key as sk=H(ID_c, M, W, M^r) shared with S. Therefore, user U's insider impersonation attack succeeds.

6. Review and attack on the protocol of Li et al.'s scheme

We first briefly review the registration phase, the login phase and the authentication phase of protocol [20] in Section 6.1, then present our attack in Section 6.2.

6.1 Review

In the registration phase, user C submits his ID_c, PW_c, and his personal biometric B_c to the server S. S issues C a smart card which stores the values of ID_c, f_c=H(B_c), and e_c=H(ID_c, x)$\oplus$H(PW_c, f_c), where x is S's secret key.

In the login phase, user C inputs ID_c and PW_c to his smart card and inputs his personal biometric B_c on the specific device to check if H(B_c) is equal to f_c stored in the smart card. If it is, the card selects a random number R_c, computes M_1= $e_c\oplus$ H(PW_c, f_c)=H(ID_c, x), M_2 = $M_1\oplus R_c$, and sends { ID_c, M_2 } to S.

In the authentication phase, after receiving { ID_c, M_2 }, S checks to see if ID_c is valid. If it is, S chooses a random number R_S, computes M_3=H(ID_c, x), M_4= $M_2\oplus M_3$= R_c, M_5 =$M_3\oplus R_S$, M_6=H(M_2, M_4), and sends { M_5, M_6} to C. After receiving S's message, C verifies whether M_6 is equal to H(M_2, R_c). If it is, S is authentic. C then computes M_7=$M_5\oplus M_1$=$M_3\oplus R_S\oplus M_1$=H(ID_c, x)$\oplus R_S\oplus$H(ID_c, x)=R_S, M_8=H(M_5, M_7), and sends {M_8} to S. After receiving C's message, S verifies whether M_8 is equal to H(M_5, R_s). If it is, C is authentic. S then accepts C's login request.

6.2 Attack

Assume that an attacker E gets C's smart card and reads the values of ID_c, f_c and e_c. He can successfully launch a password-guessing attack as shown below. E chooses a random number M_e and sends {ID_c, M_e} to S. After receiving the message, S checks to see if ID_c is valid. If it is, S chooses a random number R_S, computes M_3=H(ID_c, x), M_4= $M_e\oplus M_3$, M_5= $M_3\oplus R_S$, M_6=H(M_e, M_4), and sends { M_5, M_6} to E. After receiving S's message, E terminates the communication, chooses a possible password PW', computes M'=H(M_e, $M_e\oplus e_c\oplus$H(PW', f_c)), and compares to see if M' is equal to M_6. If they are, PW' is the correct password, since $M_e\oplus e_c\oplus$H(PW', f_c)=$M_e\oplus$H(ID_c, x)$\oplus$ H(PW_c, f_c)$\oplus$H(PW', f_c). If PW' =PW_c, then the equation equals to $M_e\oplus$H(ID_c, x) which equals to $M_e\oplus M_3$= M_4. That is, M'=H(M_e, M_4)=M_6. E can therefore masquerade as C to log into the server. In other words, the attacker can successfully implement the password guessing attack if the smart card is lost.

7. Conclusion

In this article, we have listed the ten requirements proposed by Liao et al. and used them to examine five recent smart card based password authentication protocols. Although each of them claims that their scheme is secure. However, after analyses, we found that the protocols of Juang et al.'s [7], Hsiang et al.'s [14], Kim et al.'s [16], and Li et al.'s [20] suffer from the password guessing attack if the smart card is lost and the protocol of Xu et al.'s [18] suffers from the insider attack.

References

[1] H. Y. Chien, C. H. Chen, "A Remote Authentication Preserving User Anonymity," *Proceedings of the 19th International Conference on Advanced Information Networking and Applications* (AINA '05), Vol.2, pp. 245-248, March 2005.

[2] I. E. Liao, C. C. Lee, M. S. Hwang, "A password authentication scheme over insecure networks", *Journal of Computer and System Sciences*, Vol. 72, No. 4, pp. 727-740, June 2006.

[3] T. H. Chen, W. B. Lee, "A new method for using hash functions to solve remote user authentication", Computers *& Electrical Engineering*, Vol. 34, No. 1, pp. 53-62, January 2008.

[4] C. S. Bindu, P. C. S. Reddy, B. Satyanarayana, "Improved remote user authentication scheme preserving user anonymity", *International Journal of Computer Science and Network Security*, Vol. 8, No. 3, pp. 62-65, March 2008.

[5] Y. Lee, J. Nam, D. Won, "Vulnerabilities in a remote agent authentication scheme using smart cards", *LNCS: AMSTA*, Vol. 4953, pp. 850-857, April 2008.

[6] W. S. Juang, S. T. Chen, H. T. Liaw, "Robust and efficient password-authenticated key agreement using smart cards", *IEEE Transactions on Industrial Electronics*, Vol. 55, No. 6, pp. 2551-2556, June2008.

[7] W. S. Juang, W. K. Nien, "Efficient password authenticated key agreement using bilinear pairings", *Mathematical and Computer Modelling*, Vol. 47, No. 11-12, pp. 1238-1245, June 2008.

[8] J. Y. Liu, A. M. Zhou, M. X. Gao, "A new mutual authentication scheme based on nonce and smart cards", *Computer Communications*, Vol. 31, No. 10, pp. 2205-2209, June 2008.

[9] T. Xiang, K. Wong, X. Liao, "Cryptanalysis of a password authentication scheme over insecure networks", *Computer and System Sciences*, Vol. 74, No. 5, pp. 657-661, August 2008.

[10] G. Yang, D. S. Wong, H. Wang, X. Deng, "Two-factor mutual authentication based on smart cards and passwords", *Journal of Computer and System Sciences*,

Vol. 74, No. 7, pp.1160-1172, November 2008.
[11] T. Goriparthi, M. L. Das, A. Saxena, "An improved bilinear pairing based remote user authentication scheme", *Computer Standards & Interfaces*, Vol. 31, No. 1, pp. 181-185, January 2009.
[12] H. S. Rhee, J. O. Kwon, D. H. Lee, "A remote user authentication scheme without using smart cards", *Computer Standards & Interfaces*, Vol. 31, No. 1, pp. 6-13, January 2009.
[13] Y. Y. Wang, J. Y. Liu, F. X. Xiao, J. Dan, "A more efficient and secure dynamic ID-based remote user authentication scheme", *Computer Communications*, Vol. 32, No. 4, pp. 583-585, March 2009.
[14] H. C. Hsiang, W. K. Shih, "Weaknesses and improvements of the Yoon–Ryu–Yoo remote user authentication scheme using smart cards", *Computer Communications*, Vol. 32, No. 4, pp. 649-652, March 2009.
[15] D. Z. Sun, J. P. Huai, J. Z. Sun, J. X. Li, "Cryptanalysis of a mutual authentication scheme based on nonce and smart cards", *Computer Communications*, Vol. 32, No. 6, pp. 1015-1017, April 2009.
[16] S. K. Kim , M. G. Chung, "More secure remote user authentication scheme", *Computer Communications*, Vol. 32, No. 6, pp. 1018-1021, April 2009.
[17] H. R. Chung, W. C. Ku, M. J. Tsaur, "Weaknesses and improvement of Wang et al.'s remote user password authentication scheme for resource-limited environments", *Computer Standards & Interfaces*, Vol. 31, No. 4, pp. 863-868, June 2009.
[18] J. Xu, W. T. Zhu, D. G. Feng, "An improved smart card based password authentication scheme with provable security", *Computer Standards & Interfaces*, Vol. 31, No. 4, pp. 723-728, June 2009.
[19] M. S. Hwang, S. K. Chong, T. Y. Chen, "DoS-resistant ID-based password authentication scheme using smart cards", *Journal of Systems and Software*, In Press, Available online 12 August 2009.
[20] C. T. Li, M. S. Hwang, "An efficient biometrics-based remote user authentication scheme using smart cards", *Journal of Network and Computer Applications*, Vol. 33, No. 1, pp. 1-5, January 2010.