Weaknesses of a Multi-Server Password Authenticated Key Agreement Scheme

一個多重伺服器架構下的通行碼身份認證與金鑰協議設計之弱點

Wei-Chi Ku (顧維祺)
Dept of Computer Sci. and Info. Engineering
Fu Jen Catholic University
輔仁大學資訊工程學系
E-mail: wcku@csie.fju.edu.tw

Hsiu-Mei Chuang (莊秀美)
Dept of Computer Sci. and Info. Engineering
Fu Jen Catholic University
輔仁大學資訊工程學系
E-mail: vivia@wcku1.csie.fju.edu.tw

Min-Hung Chiang (江旻紘)
Dept of Computer Sci. and Info. Engineering
Fu Jen Catholic University
輔仁大學資訊工程學系
E-mail: grievous@wcku1.csie.fju.edu.tw

Kuo-Tsai Chang (張國財)
Dept of Computer Sci. and Info. Engineering
Fu Jen Catholic University
輔仁大學資訊工程學系
E-mail: jovic@wcku1.csie.fju.edu.tw

摘要

最近，Juang 在多重伺服器的架構下提出了一個使用智慧卡的通行碼身份認證與金鑰協議設計，並宣稱其設計可以提供雙向身份認證與交談金鑰協議服務。在本文中，我們將指出 Juang 的設計仍無法抵擋內部特權者攻擊且系統可回復性較差，此外，Juang 的設計亦未能提供 forward secrecy。

關鍵詞：金鑰協議、多重伺服器架構、雙向身份認證、通行碼、智慧卡。

Abstract

Recently, Juang proposed an efficient password authenticated key agreement scheme using smart cards for the multi-server architecture, and claimed that his scheme was intended to provide mutual authentication and session key agreement. In this paper, we show that Juang's scheme is still vulnerable to a privileged insider's attack and is not reparable. Furthermore, it does not provide forward secrecy.

Keywords: key agreement, multi-server architecture, mutual authentication, password, smart card.

1. Introduction

A common feature of conventional password authentication schemes is that a verification table, which contains the verifiers of users' passwords, should be securely stored in the server. If the verifier is stolen or modified by the adversary, the system will be breached. In 1990, Hwang, Chen, and Laih [5] initially proposed a non-interactive password authentication scheme and its enhanced version, which additionally uses smart cards. Their schemes are novel because the server does not require storing verifiers and the server does not need to keep any secret of the user. However, Hwang-Chen-Laih's enhanced scheme still has several drawbacks and weaknesses, e.g., passwords are difficult to memorize, and users can not freely choose and change passwords. Since then, many verifier-free password authentication schemes using smart cards have been proposed, e.g., [1]–[3], [7], [8], [10], [13], [16]–[19], and each has its pros and cons. However, all these schemes are designed for the single-server architecture. If there are multiple servers to access, the user has to register with each server individually and possibly should remember different identifications and passwords for accessing different servers.

In 2001, Li, Lin, and Hwang [14] described a verifier-free password authentication scheme for the multi-server architecture by using neural networks. Their scheme has the merit that the user does not need to individually register with each server. However, Li-Lin-Hwang's scheme is inefficient for large-scale environments because it spends too much time training neural networks. In 2003, Lin, Hwang, and Li [12] proposed an efficient verifier-free password authentication scheme using smart cards for the multi-server architecture based on the geometric property of the Euclidean plane, and claimed that their scheme is secure against the replay attack, the forgery attack, the guessing attack, and the modification attack. In addition, the user can freely choose/change password. However, Lin-Hwang-Li's scheme does not provide mutual authentication and session key agreement, and thus its application is restricted.

Recently, Juang [9] proposed an efficient password authenticated key agreement scheme using smart cards for the multi-server architecture. The merits of Juang's scheme are: (1) the user only has to register with the registration center once and can access all the servers within the system; (2) no verification table or password table is stored in the server; (3) the user can freely choose password; (4) the computation and communication cost is low; (5) the user and the server can authenticate each other; (6) a session key is established between the user and the server for each session; and (7) system clock synchronization is not required. Unfortunately, we find that Juang's scheme is vulnerable to a privileged insider's attack and is not reparable [6]. Additionally, Juang's scheme does not provide forward secrecy [4]. In this paper, we will describe the weaknesses of Juang's scheme.

2. Review of Juang's Scheme

In the multi-server architecture of Juang's scheme [9], there are three kinds of participants: users, servers, and a registration center. The user only has to register with the registration center once and then can obtain the services from a set of servers, i.e., the user does not need to individually register with each of these servers. The registration center is responsible for setting up several public/secret parameters and publishing some system information. The notations used throughout this paper are summarized in Table 1.

Table 1. Notations of Juang's scheme

Notation	Description
RC	the registration center
U_i	the user i
S_j	the server j
UID_i	the unique identification of U_i
SID_j	the unique identification of S_j
PW_i	the password of U_i
x	the secret key secretly selected and kept by RC
$E_k(\cdot)$	the encryption function of a symmetric cryptosystem with secret key k
$D_k(\cdot)$	the decryption function corresponding to $E_k(\cdot)$
$h(\cdot)$	a secure one-way hash function
$\oplus$	the bitwise exclusive-or operation
$\Vert$	the string concatenation operator
'$A \rightarrow B : m$'	A sends m to B through a common communication channel

Initially, for each server, say S_j, RC computes $w_j = h(x, SID_j)$ and then sends w_j to S_j through a secure channel. The secret key w_j is securely shared between RC and S_j. The scheme involves the registration phase, the login and session key agreement phase, and the shared key inquiry phase, which can be described as in the following.

Registration Phase

This phase is invoked when U_i requests to register with RC.

Step R1. U_i submits UID_i and PW_i to RC for registration.

Step R2. RC computes

$$v_i = h(x, UID_i)$$

$$\mu_i = v_i \oplus PW_i.$$

Step R3. RC delivers a smart card containing UID_i and μ_i to U_i through a secure channel.

Step R4. For each server, say S_j, RC computes

$$v_{i,j} = h(v_i, SID_j)$$

$$a_{i,j} = Ew_j(v_{i,j}, UID_i)$$

and sends $a_{i,j}$ to S_j. Then, S_j can choose to either store $a_{i,j}$ in his encrypted keys table or ignore it according to whether he has maintained an encrypted keys table or not.

Login and Session Key Agreement Phase

This phase is invoked whenever U_i requests to login S_j.

Step L1. U_i inserts his smart card into the smart card reader of a terminal, and then enters UID_i and PW_i into his smart card. Next, U_i's smart card generates two random values r_u and N_1, where r_u is used for generating the session key and N_1 is used as U_i's nonce, and then computes

$$v_i = \mu_i \oplus PW_i$$

$$v_{i,j} = h(v_i, SID_j)$$

$$c_1 = E_{v_{i,j}}(r_u, h(UID_i \| N_1)).$$

Step L2. $U_i \rightarrow S_j : N_1, UID_i, c_1$.

Step L3. If S_j has not maintained an encrypted keys table, the shared key inquiry phase is invoked. Otherwise, S_j retrieves $a_{i,j} = E_{w_j}(v_{i,j}, UID_i)$ from his encrypted keys table and computes $D_{w_j}(a_{i,j})$ to derive $v_{i,j}$ and UID_i. Then, S_j uses $v_{i,j}$ to compute $D_{v_{i,j}}(c_1)$, which yields r_u and $h(UID_i \| N_1)$. In addition, S_j uses UID_i and N_1 to compute $h(UID_i \| N_1)$. If the computed $h(UID_i \| N_1)$ equals the decrypted one and N_1 is fresh, S_j generates two random values r_s and N_2, where r_s is used for generating the session key and N_2 is used as S_j's nonce. Next, S_j computes

$$sk = h(r_s, r_u, v_{i,j})$$

$$c_2 = E_{v_{i,j}}(r_s, N_1+1, N_2),$$

where sk is used as the session key between U_i and S_j.

Step L4. $S_j \rightarrow U_i : c_2$.

Step L5. U_i's smart card computes $D_{v_{i,j}}(c_2)$. If the second decrypted item equals the expected N_1+1, U_i's smart card computes

$$sk = h(r_s, r_u, v_{i,j})$$

$$c_3 = E_{sk}(N_2+1).$$

Step L6. $U_i \rightarrow S_j : c_3$.

Step L7. S_j computes $D_{sk}(c_3)$, and if the decrypted item equals the expected N_2+1, S_j successfully authenticates U_i. Then, S_j and U_i can use sk to secure subsequent messages exchanged in this session.

Shared Key Inquiry Phase

This phase is invoked in the beginning of Step L3 in the case that S_j has not maintained an encrypted keys table.

Step S1. S_j generates a random value N_3, which is used as S_j's nonce, and then computes $h(UID_i \| SID_j \| N_3)$ and $c_4 = E_{w_j}(h(UID_i \| SID_j \| N_3))$.

Step S2. $S_j \rightarrow RC : N_3, UID_i, SID_j, c_4$.

Step S3. Upon receiving S_j's shared key inquiry message, RC computes $D_{w_j}(c_4)$ to derive $h(UID_i \| SID_j \| N_3)$, and uses the received N_3, UID_i, and SID_j to compute $h(UID_i \| SID_j \| N_3)$. If the computed $h(UID_i \| SID_j \| N_3)$ equals the decrypted one and N_3 is fresh, RC computes

$$v_{i,j} = h(v_i, SID_j)$$

$$c_5 = E_{w_j}(v_{i,j}, N_3+1).$$

Step S4. $RC \rightarrow S_j : c_5$.

Step S5. S_j computes $D_{w_j}(c_5)$ to derive $v_{i,j}$ and N_3+1. If the second decrypted item equals the expected N_3+1, S_j authenticates $v_{i,j}$. Next, Step L3 is resumed.

3. Weaknesses of Juang's Scheme

In this section, we will show the weaknesses of Juang's scheme [9].

Poor Reparability

Although the tamper resistance of smart cards was widely assumed in their applications, such an assumption may be problematic in practice. Many researches have demonstrated that the secrets stored in a smart card can be breached by monitoring the power consumption, e.g., [11], or analyzing the leaked information, e.g., [15]. Suppose that the adversary has obtained the μ_i stored in U_i's smart card and also has intercepted the message transmitted in Step L2, i.e., $\{N_1, UID_i, c_1\}$, during one of U_i's past logins. Then, the adversary can guess a candidate password PW_i' and compute

$$v_i' = \mu_i \oplus PW_i'$$

$$v_{i,j}' = h(v_i', SID_j)$$

$$r_u', h(UID_i \| N_1)'\} = D_{v_{i,j}'}(c_1).$$

Next, the adversary computes $h(UID_i \| N_1)$ and compares the result to $h(UID_i \| N_1)'$. If they are equal, the adversary has obtained $v_{i,j}' = v_{i,j}$, which also implies that he has obtained $v_i' = v_i$ and $PW_i' = PW_i$. Otherwise, the adversary tries another candidate password. After obtaining v_i, the adversary can generate $v_{i,k} = h(v_i, SID_k)$ for any k such that S_k is within the system, and then use $v_{i,k}$ to impersonate U_i to login S_k or impersonate S_k to fool U_i. Additionally, the adversary can use $v_{i,k}$ to perform a man-in-the-middle attack by establishing parallel sessions with U_i and S_k, respectively. Unfortunately, the above described impersonation attack and man-in-the-middle attack can not be stopped even if U_i has detected that v_i has been compromised and then used a new password to re-register with *RC*. As the value of v_i is unrelated to U_i's password and instead is determined only by U_i's identification UID_i and *RC*'s permanent secret key x, *RC* can not change v_i for U_i unless UID_i or x can be changed. However, since x is commonly used for all users rather than specifically used for only U_i, it is unreasonable and inefficient if x should be changed to recover the security of U_i only. In addition, it is also impractical to change UID_i, which should be tied to U_i in most application systems. Hence, Juang's scheme is not reparable [6].

Lack of Forward Secrecy

Suppose that $v_{i,j}$, which is shared by U_i and S_j, has been compromised by the adversary. As previously described, the adversary can impersonate U_i to login S_j or impersonate S_j to fool U_i. Furthermore, we will show that the adversary can derive the session key used in any previous session between U_i and S_j as follows. By using $v_{i,j}$ to decrypt c_1' ($= E_{v_{i,j}}(r_u', h(UID_i \| N_1'))$), which was intercepted in Step L2 of any previous session, the adversary can obtain r_u'. Similarly, by using $v_{i,j}$ to decrypt c_2' ($= E_{v_{i,j}}(r_s', N_1'+1, N_2')$), which was intercepted in Step L4 of the corresponding session, the adversary can obtain r_s'. Next, the aversary can compute the session key $sk' = h(r_u', r_s', v_{i,j})$, and then use sk' to decrypt all the messages exchanged between U_i and S_j in the corresponding session. Therefore, Juang's scheme fails to provide forward secrecy [4]. Note that if Diffie-Hellman key exchange scheme is employed in establishing the session key to achieve forward secrecy, the expected advantages of Juang's scheme over similar schemes with respect to computation overhead and implementation cost vanish.

Vulnerability to Privileged Insider's Attack

In practice, it is likely that the user uses the same password to access several servers for his convenience. In Step R1 of the registration phase, U_i's password PW_i will be revealed to *RC*. Then, the privileged insider of *RC* may try to use PW_i to impersonate U_i to access the servers outside this system. If the targeted outside server adopts the normal password authentication scheme, it is possible that the privileged insider of *RC* can successfully impersonate U_i to login it by using PW_i. Although it is also possible that all the privileged insiders of *RC* are trusted and U_i does not use the same password to access several servers, the implementers and the users of the scheme should be aware of such a potential weakness. For this reason, in many password authentication schemes, e.g., [1], [8], [10], [13], [17], the user's password is not revealed to others including the registration center and the servers.

4. Misleading Claims

Next, we will address the misleading security related claims made in Juang's scheme. In Step L3 of the login and session key agreement phase, it is claimed that S_j can verify the freshness of the N_1 received in Step L2. However, since S_j has not recorded all the nonces received from U_i, he can not judge whether N_1 is fresh or not. Actually, S_j can only be assured after successfully verifying c_1 ($= E_{v_{i,j}}(ru, h(UID_i \| N_1))$) that N_1 is or was ever sent by U_i. Similarly, the claim made in Step S3 that *RC* can verify the freshness of the N_3 received in Step S2 is also inappropriate. It should be noted that these two wrong claims may be employed by the adversary to carry out some subtle attacks to the application systems.

5. Conclusion

Juang's verifier-free password authentication scheme using smart cards for the multi-server architecture is novel and interesting in that it additionally provides mutual authentication and key agreement. In comparison with similar schemes, the involved computation and communication cost of Juang's scheme is low. However, the security strength of Juang's scheme is not ideal enough. In this paper, we have demonstrated that Juang's scheme is vulnerable to a privileged insider's attack and is not reparable. Furthermore, Juang's scheme does not provide forward 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] H. Y. Chien, J. K. Jan, and Y. M. Tseng, "A modified remote login authentication scheme based on geometric approach," The Journal of Systems and Software, vol. 55, no. 3, pp. 287-290, Jan. 2001.

[2] H. Y. Chien, J. K. Jan, and Y. M. Tseng, "An efficient and practical solution to remote authentication: smart card," Computers & Security, vol. 21, no. 4, pp. 372-375, Aug. 2002.

[3] C. C. Chang and T. C. Wu, "Remote password

authentication with smart cards," IEE Proceedings-E, vol. 138, no. 3, pp. 165-168, May 1991.

[4] W. Diffie, P. C. van Oorschot, and M. J. Wiener, "Authentication and authenticated key exchanges," Designs, Codes and Cryptography, vol. 2, no. 2, pp. 107-125, June 1992.

[5] T. Hwang, Y. Chen, and C. S. Laih, "Non-interactive password authentications without password tables," IEEE Region 10 Conference on Computer and Communication Systems, Hong Kong, pp. 429-431, Sept. 1990.

[6] T. Hwang and W. C. Ku, "Reparable key distribution protocols for Internet environments," IEEE Trans. Commun., vol. 43, no. 5, pp. 1947-1949, May 1995.

[7] M. S. Hwang and L. H. Li, "A new remote user authentication scheme using smart card," IEEE Trans. Consumer Electron., vol. 46, no. 1, pp. 28-30, Feb. 2000.

[8] M. S. Hwang, C. C. Lee, and Y. L. Tang, "A simple remote user authentication scheme," Mathematical and Comput. Modelling, vol. 36, no. 1-2, pp. 103-107, July 2002.

[9] W. S. Juang, "Efficient multi-server password authenticated key agreement using smart cards," IEEE Trans. Consumer Electron., vol. 50, no. 1, pp. 251-255, Feb. 2004.

[10] W. C. Ku and S. M. Chen, "Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards," IEEE Trans. Consumer Electron., vol. 50, no. 1, pp. 204-207, Feb. 2004.

[11] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," Proc. Advances in Cryptology (CRYPTO'99), pp. 388-397, 1999.

[12] I. C. Lin, M. S. Hwang, and L. H. Li, "A new remote user authentication scheme for multi-server architecture," Future Generation Comput. Systems, vol. 19, no. 1, pp. 13-22, Jan. 2003.

[13] C. C. Lee, M. S. Hwang, and W. P. Yang, "A flexible remote user authentication scheme using smart cards," ACM Operat. Syst. Rev., vol. 36, no. 3, pp. 46-52, July 2002.

[14] L. H. Li, I. C. Lin, and M. S. Hwang, "A remote password authentication scheme for multiserver architecture using neural networks," IEEE Trans. Neural Networks, vol. 12, no. 6, pp. 1498-1504, Nov. 2001.

[15] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," IEEE Trans. Comput., vol. 51, no. 5, pp. 541-552, May 2002.

[16] H. M. Sun, "An efficient remote user authentication scheme using smart cards," IEEE Trans. Consumer Electron., vol. 46, no. 4, pp. 958-961, Nov. 2000.

[17] T. C. Wu, "Remote login authentication scheme based on a geometric approach," Comput. Commun., vol. 18, no. 12, pp. 959-963, Dec. 1995.

[18] S. J. Wang and J. F. Chang, "Smart card based secure password authentication scheme," Computers & Security, vol. 15, no. 3, pp. 231-237, 1996.

[19] W. H. Yang and S. P. Shieh, "Password authentication schemes with smart cards," Computers & Security, vol. 18, no. 8, pp. 727-733, 1999.