AN EFFICIENT AND PRACTICAL REMOTE USER AUTHENTICATION SCHEME

Ya-Fen CHANG and Chin-Chen CHANG

Abstract: In 2000, Peyravian and Zunic proposed a simple and efficient password authentication scheme based on the collision-resistant hash function. Later, Hwang and Yeh indicated that Peyravian and Zunic's scheme is insecure and proposed an improvement by using the server's public key. Nevertheless, in practice, services that do not use public keys are quite often superior to PKIs. At the same time, Lee, Li and Hwang indicated that Peyravian and Zunic's scheme suffers from off-line password guessing attacks and presented an improved version. However, Lee-Li-Hwang's proposed scheme is still vulnerable to the same attacks and denial-of-service attacks. Therefore, this article presents a secure and efficient improvement.

Keywords: Password Authentication, Password Guessing Attacks.

Introduction

Several authentication methods have been proposed for electronic commerce environments (e.g., the authentication service Kerberos[1]). Among them, the password authentication scheme is the most common approach. In the password authentication scheme, a client is allowed to share an easy-to-remember password with a trusted server. Such concepts are applied in other applications as well.[2,3,4] Due to the characteristic of easy-to-remember password, these schemes may be broken with a little effort. Ding and Horster[5] divide the password-guessing attacks into three classes: (1) detectable on-line password guessing attacks, (2) undetectable on-line password guessing attacks, and (3) off-line password guessing attacks. In 2000, Peyravian and Zunic[6] proposed a novel password authentication scheme. The proposed scheme employs a collision-resistant hash function to protect the transmission of the password over an insecure network. No symmetric-key or public-key based authentication system is required in their method. Instead, only the hash value of the password is transmitted. In addition, random numbers are used to avoid eavesdropping and replay.

In 2002, Hwang and Yeh[7] pointed out that the security of Peyravian and Zunic's pass-

word authentication scheme is only based on the user password. Because of the features of the easy-to-remember passwords and the fact that no additional authentication approach is used, they pointed out that Peyravian and Zunic's password authentication schemes cannot resist password guessing attacks, server spoofing, and server data eavesdropping. As a result, they proposed an improvement to defend against the above-mentioned three types of attacks by using the server's public key. Notwithstanding, applying the server's public key puts a high burden on users since the server's public key needs to be verified before being used. In fact, in practice services that do not use public keys are quite often superior to PKIs.

At the same time, Lee, Li, and Hwang[8] also found the security flaws in the scheme proposed by Peyravian and Zunic. They presented another improvement and claimed that their scheme has the same features, such that still only the collision-resistant hash function is used to protect the transmission of the password. However, the Lee-Li-Hwang's scheme is still vulnerable to off-line password guessing attacks. Moreover, the authorized user will suffer from the denial-of-service attacks while changing his/her password. Consequently, the authors of this article propose a secure and efficient improved scheme where the computational load is low and the server's public key is not used.

This article is organized as follows. The next section reviews the scheme proposed by Hwang and Yeh. Then, the authors provide a review and cryptanalysis of the Lee-Li-Hwang's scheme. A novel improved scheme is presented afterwards, followed by a comprehensive discussion. Conclusions are drawn in the last section.

Review of Hwang and Yeh's Scheme

This section presents Hwang and Yeh's proposed improvement on Peyravian-Zunic's password authentication scheme. The initiation goes as follows. The client U_i with identity d_i and the trusted server S share a secret password p_i. S has a server public key KS. $H(p_i)$ is stored in the database by S for U_i, where $H()$ is a collision-resistant hash function. $EK()$ denotes an asymmetric encryption scheme with a public key K.

Hwang and Yeh's password authentication scheme consists of two phases: a password authentication phase and a password change phase. The password authentication phase is shown in Figure 1. The details are as follows:

Step1. U_i computes and sends $E_{K_S}(r_C, p_i)$ to S, where r_C is a random number, with d_i to S as a login request.

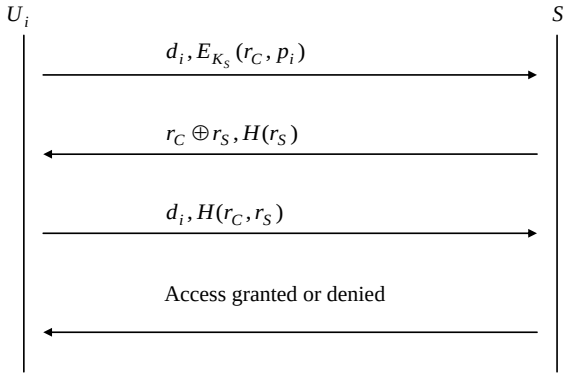

Figure 1: The Password Authentication Phase of Huang and Yeh's Scheme.

Step 2. After receiving the login request sent from U_i, S uses his/her own private key to retrieve r_C and p_i. Then S computes $H(p_i)$ for comparison with the corresponding item stored in the database. In case of equal values, S computes and sends $r_C \oplus r_S$ and $H(r_S)$ to U_i, where r_S is a random number chosen by S and $\oplus$ denotes XOR; otherwise, S terminates the protocol.

Step 3. Upon receiving the transmitted data, U_i computes $H(r_C \oplus (r_C \oplus r_S))$ and checks whether the result of this computation and $H(r_S)$ are equal. If it holds, U_i computes and sends $H(r_C, r_S)$ with d_i to S; otherwise, U_i may restart the scheme or ask S to retransmit the necessary information.

Step 4. After getting $(d_i, H(r_C, r_S))$, S computes and compares $H(r_C, r_S)$ with the received message. If they are equal, S grants U_i access request. Otherwise, S rejects the access request from U_i.

In the password change phase, the steps are almost the same as those of the password authentication scheme, except for an additional password change request in Step 3. In Step 3, U_i sends $d_i, H(r_C, r_S), H(p_i') \oplus H(r_C + 1, r_S)$ and the password change request to S, where p_i' is the new password chosen by U_i. Then, S computes $H(r_C + 1, r_S)$ and $H(r_C + 1, r_S) \oplus (H(p_i') \oplus H(r_C + 1, r_S))$ to get U_i's new verifier $H(p_i')$.

Review and Cryptanalysis of Lee-Li-Hwang's Scheme

First, the authors review the scheme proposed by Lee, Li and Hwang, followed by its cryptanalysis.

Review of Lee-Li-Hwang's Scheme

As stated in the previous section, U_i with identity d_i and the trusted server S share a secret password p_i. S stores $HPW_i = H(d_i, p_i)$ for U_i, where $H()$ is a collision-resistant hash function. The password authentication phase is shown in Figure 2. The details follow:

Step 1. U_i chooses a random number r_C. Then s/he computes and sends $r_C \oplus HPW_i$ with d_i to S as a login request, where $\oplus$ denotes XOR.

Step 2. After receiving the login request, S computes $HPW_i \oplus (r_C \oplus HPW_i)$. Then S computes and sends $r_S \oplus HPW_i$ to U_i, where r_S is a random number.

Step 3. Upon receiving the transmitted data, U_i computes $r_S = HPW_i \oplus (r_S \oplus HPW_i)$ and $AUTH = H(HPW_i, r_C, r_S)$. Then U_i sends d_i and $AUTH$ to S.

Step 4. After getting d_i and $AUTH$ from U_i, S computes and compares $H(HPW_i, r_C, r_S)$ with $AUTH$. If they are equal, S grants U_i the access request. Otherwise, S rejects the access request from U_i.

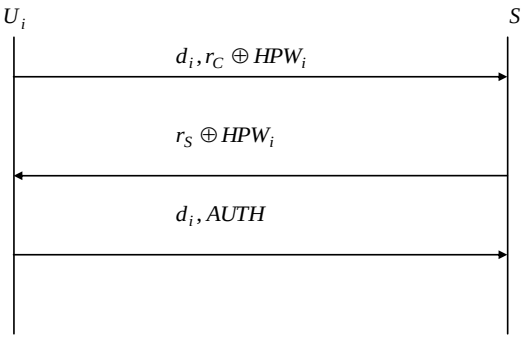

Figure 2: The Password Authentication Phase of Lee-Li-Hwang's Scheme.

The password changing phase of Lee-Li-Hwang's scheme follows the procedure illustrated in Figure 3. The steps are very similar to those of the password authentication phase, except for an additional password change request in Step 3. In Step 3, U_i computes $HPW_i' = H(d_i, p_i')$ and $Mask = HPW_i' \oplus H(HPW_i, r_C + 1, r_S)$, where p_i' is the new password chosen by U_i. Then U_i sends d_i, $AUTH$, and $Mask$ and the password change request to S. After getting the password change request, S first

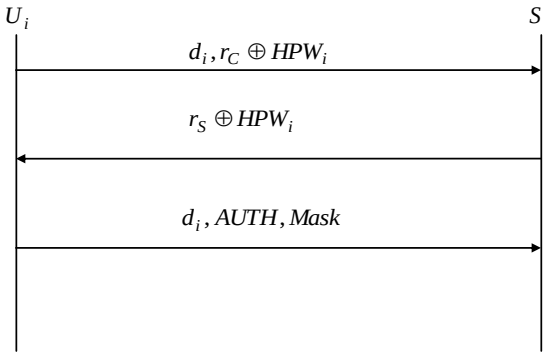

Figure 3: The Password Change Phase of Lee-Li-Hwang's Scheme.

authenticates U_i by comparing $H(HPW_i, r_C, r_S)$ with $AUTH$. If it does not hold, S denies the password change request; otherwise, S computes $HPW_i' = H(HPW_i, r_C + 1, r_S) \oplus Mask$ and updates U_i's verifier with HPW_i'.

Cryptanalysis of Lee-Li-Hwang's Scheme

Lee, Li and Hwang claimed that the scheme they propose can resist off-line password guessing attacks since r_C and r_S are random and the transmitted verifier is concealed. However, the authors will show that Lee-Li-Hwang's password authentication scheme still suffers from off-line password guessing. Even more, the proposed scheme cannot resist denial-of-service attacks when the authorized user changes his/her password. How to mount denial-of-service attacks on Lee-Li-Hwang's proposed scheme is shown afterwards.

Security Flaw 1: Lee-Li-Hwang's scheme suffers from off-line password guessing attacks

Consider the scenario when a malicious user Eve eavesdrops the data transmitted between U_i and S. As a result, Eve knows $r_C \oplus HPW_i$, $r_S \oplus HPW_i$, and $AUTH = H(HPW_i, r_C, r_S)$. Then, Eve performs the following operations:

Step 1. Eve guesses U_i's password to be pw and computes $HPW = H(d_i, pw)$.

Step 2. Eve computes $r_C'' = (r_C \oplus HPW_i) \oplus HPW$, $r_S'' = (r_S \oplus HPW_i) \oplus HPW$, and $AUTH'' = H(HPW_i, r_C'', r_S'')$.

Step 3. Eve checks whether $AUTH = AUTH''$ holds or not. If it holds, Eve is sure that pw is p_i. Otherwise, Eve repeats Steps 1 to 3.

According to the procedure described above, it is obvious that Eve can successfully get U_i's password p_i by performing off-line password guessing attacks.

Security Flaw 2: Lee-Li-Hwang's scheme suffers from denial-of-service attacks when the authorized user wants to change the password

When U_i wants to change her/his password, s/he follows the procedure as shown in Figure 3. Eve intercepts the transmitted data (d_i, $AUTH$, $Mask$) in Step 3 and replaces $Mask$ with a random number R, where $|R| = |Mask|$. Then Eve sends d_i, $AUTH$, and R and the password change request to S. After getting the request, S first authenticates U_i by comparing $H(HPW_i, r_C, r_S)$ with $AUTH$. It is obvious that U_i will be authenticated successfully since U_i indeed knows p_i. Then S computes $HPW_i'' = H(HPW_i, r_C + 1, r_S) \oplus R \neq HPW_i'$ and updates U_i's verifier with HPW_i''. Later, when U_i wants to access S, U_i will not be authenticated successfully.

An Efficient and Practical Remote User Authentication Scheme

In this section, the authors propose their novel remote user authentication scheme. To start with, the server S and the user U_i share a password p_i. There is one public system parameter $n = p \times q$, where p and q are two secret system parameters known only to S. S stores $(x \oplus p_i)$ in the database, where x is a system parameter kept concealed by S. The proposed scheme consists of two phases: a password authentication phase and a password change phase, which are described below.

Remote Password Authentication Phase

The password authentication phase is shown in Figure 4, and its description is given below:

Step 1. U_i sends access request containing her/his identity d_i and the timestamp.

Step 2. S uses x to retrieve p_i by computing $x \oplus (x \oplus p_i)$ and calculates $E1_{p_i}(r_S)$, where $E1()$ is a symmetric encryption algorithm and r_S is a random number. Then S sends $E1_{p_i}(r_S)$ to U_i.

Step 3. After getting the transmitted data, U_i computes $r_S = D1_{p_i}(E1_{p_i}(r_S))$, where $D1()$ is a symmetric decryption algorithm. Then U_i chooses $s_i \in_R Z_n$ and calculates $\alpha = H(r_S, s_i)$ and $z = s_i^2 \bmod n$. After that, U_i sends α and z to S.

Step 4. After getting α and z, S computes

$a_1 = z^{(p+1)/4} \bmod p$,

$a_2 = (p - z^{(p+1)/4}) \bmod p$,

$a_3 = z^{(q+1)/4} \bmod q$,

$a_4 = (q - z^{(q+1)/4}) \bmod q$,

$x = q(q^{-1} \bmod p)$, $y = p(p^{-1} \bmod q)$,

$\beta_1 = (x * a_1 + y * a_3) \bmod n$,

$\beta_2 = (x * a_1 + y * a_4) \bmod n$,

$\beta_3 = (x * a_2 + y * a_3) \bmod n$, and

$\beta_4 = (x * a_2 + y * a_4) \bmod n$.

For j = 1, 2, 3, and 4 let $s_i' = \beta_j$. S computes $\alpha' = H(r_S, s_i')$. Then S checks whether any α' and α are equivalent. If it does not hold, S terminates the protocol; otherwise, S accepts the access request and sends $H(d_i, s_i')$ to U_i.

Step 5. After getting $H(d_i, s_i')$ from S, U_i checks whether $H(d_i, s_i')$ and $H(d_i, s_i)$ are equivalent. If it holds, U_i is convinced that the communication party is indeed S.

Remote Password Change Phase

When U_i wants to update p_i with the new password p_i', the remote password change phase will look like as shown in Figure 5. The algorithm of the remote password change phase is as follows:

Step 1. U_i sends a password-change request containing her/his identity d_i and the timestamp.

Step 2. S computes $x \oplus (x \oplus p_i)$ to retrieve p_i and calculates $E1_{p_i}(r_S)$. Then S sends $E1_{p_i}(r_S)$ to U_i.

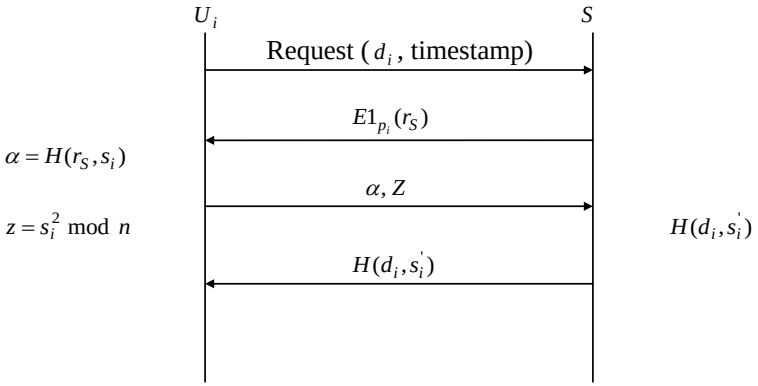

Figure 4: Proposed Remote Password Authentication Phase.

Step 3. After getting the transmitted data, U_i computes

$$r_S = D1_{p_i}(E1_{p_i}(r_S)).$$

Then U_i chooses the new password p_i' and calculates

$$\alpha = H(r_S, s_i),$$

$$Check = E1_{s_i}(p_i', r_S), \text{ and}$$

$$z = s_i^2 \bmod n.$$

Afterwards, U_i sends α, *Check* and z to S.

Step 4. After getting α, *Check* and z, S computes

$$a_1 = z^{(p+1)/4} \bmod p,$$

$$a_2 = (p - z^{(p+1)/4}) \bmod p,$$

$$a_3 = z^{(q+1)/4} \bmod q,$$

$$a_4 = (q - z^{(q+1)/4}) \bmod q,$$

$$x = q(q^{-1} \bmod p),\ y = p(p^{-1} \bmod q),$$

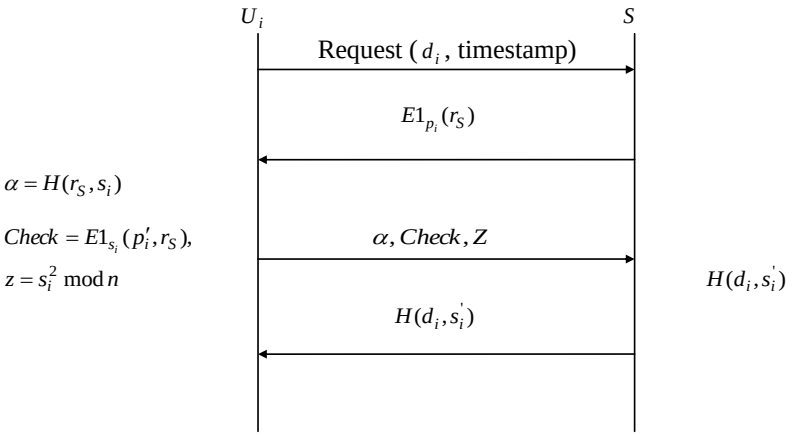

Figure 5: The Proposed Remote Password Change Phase.

$\beta_1 = (x * a_1 + y * a_3) \bmod n$,

$\beta_2 = (x * a_1 + y * a_4) \bmod n$,

$\beta_3 = (x * a_2 + y * a_3) \bmod n$, and

$\beta_4 = (x * a_2 + y * a_4) \bmod n$.

For j =1, 2, 3, and 4 let $s_i' = \beta_j$. S computes $\alpha' = H(r_S, s_i')$. Then S checks whether any α' and α are equivalent. If it does not hold, S terminates the protocol; otherwise, S computes $D1_{s_i}(Check)$ and checks whether the computation result contains r_S'. If it holds, S is convinced that p_i' is the new password and updates $(x \oplus p_i)$ with $(x \oplus p_i')$. From now on, S and U_i share the password p_i'. Then S computes and sends $H(d_i, s_i')$ to U_i.

Step 5. After getting $H(d_i, s_i')$ from S, U_i checks whether $H(d_i, s_i')$ and $H(d_i, s_i)$ are equal. If it holds, U_i is convinced that the communication party is indeed S.

Discussion

In this section, it will be shown that the proposed scheme is secure and efficient. First, security analysis is provided. Then, it will be demonstrated that the proposed

scheme is efficient by presenting performance analyses.

Security Analysis

Here, it will be demonstrated that the proposed scheme is secure. Moreover, the security drawbacks found in Lee-Li-Hwang's scheme are overcome.

Property 1: The proposed scheme can defend against the password guessing attacks

In Step 4 of the password authentication phase and the remote password change phase, S computes $a_1 = z^{(p+1)/4} \bmod p$, $a_2 = (p - z^{(p+1)/4}) \bmod p$, $a_3 = z^{(q+1)/4} \bmod q$, $a_4 = (q - z^{(q+1)/4}) \bmod q$, $x = q(q^{-1} \bmod p)$, $y = p(p^{-1} \bmod q)$, $\beta_1 = (x * a_1 + y * a_3) \bmod n$, $\beta_2 = (x * a_1 + y * a_4) \bmod n$, $\beta_3 = (x * a_2 + y * a_3) \bmod n$, and $\beta_4 = (x * a_2 + y * a_4) \bmod n$. For j = 1, 2, 3, and 4 let $s_i' = \beta_j$. S computes $\alpha' = H(r_S, s_i')$. Then S checks whether any α' and α are equivalent. If it does not hold, S will notice that there is the possibility that someone mounts attacks on the protocol. That is, it is impossible for undetectable on-line password guessing attacks to appear in the proposed protocol, and the proposed protocol is secure enough to defeat detectable on-line password guessing attacks.

With deep insight into the off-line password guessing attacks, meaningful information encrypted by the password will result in damage. The reason is that the attacker can guess the password and can decrypt the encrypted information by the guessed password. If the decryption result contains the meaningful information, it denotes that the attacker has already gotten the right password. In the proposed protocol, U_i's password is only involved in computing $E1_{p_i}(r_S)$, where r_S is a random number. No meaningful information, such as the identity or the timestamp, is contained in $E1_{p_i}(r_S)$. Hence, an attacker cannot determine whether the guessed password is correct according to $E1_{p_i}(r_S)$. In Step 3, U_i will send $\alpha = H(r_S, s_i)$ and $z = s_i^2 \bmod n$ to S. An attacker still cannot determine whether the guessed password is correct because of the following reasons: (1) $s_i \in_R Z_n$, (2) it is impossible to factor n to retrieve s_i, (3) it is impossible to know (r_S, s_i) from α for comparison since $H()$ is a one-way function. Due to the above reasons, we can sum up that the proposed scheme is secure to resist the password guessing attacks.

Property 2: The proposed scheme ensures mutual authentication

In Step 4 of both the password authentication phase and the remote password change phase, S authenticates U_i. In Step 5, U_i authenticates S by checking whether

$H(d_i, s_i')$ and $H(d_i, s_i)$ are equal. Consequently, mutual authentication is preserved in the proposed scheme. That is, server spoofing attacks cannot work in the proposed scheme.

Property 3: The proposed scheme can resist the denial-of-service attacks

In Step 3 of the remote password change phase, U_i chooses the new password p_i' and calculates $\alpha = H(r_S, s_i)$, $Check = E1_{s_i}(p_i', r_S)$, and $z = s_i^2 \bmod n$. Then, U_i sends α, $Check$ and z to S. In Step 4, S computes $a_1 = z^{(p+1)/4} \bmod p$, $a_2 = (p - z^{(p+1)/4}) \bmod p$, $a_3 = z^{(q+1)/4} \bmod q$, $a_4 = (q - z^{(q+1)/4}) \bmod q$, $x = q(q^{-1} \bmod p)$, $y = p(p^{-1} \bmod q)$, $\beta_1 = (x * a_1 + y * a_3) \bmod n$, $\beta_2 = (x * a_1 + y * a_4) \bmod n$, $\beta_3 = (x * a_2 + y * a_3) \bmod n$, and $\beta_4 = (x * a_2 + y * a_4) \bmod n$. For j =1, 2, 3, and 4 let $s_i' = \beta_j$. S computes $\alpha' = H(r_S, s_i')$. Then S checks whether any α' and α are equivalent. If it holds, S uses s_i' as the secret key to compute and checks whether the computation result contains r_S'. If it holds, S is convinced that p_i' is the new password and updates $(x \oplus p_i)$ with $(x \oplus p_i')$. The proposed approach first ensures that U_i is the authorized user and, second, makes sure that the new password is indeed the one chosen by U_i. Even though an attacker retransmits $Check$ from other iteration, S will not be cheated since s_i is a one-time used random number. Moreover, mutual authentication is confirmed in the proposed scheme. Furthermore, U_i can check whether the password is updated in Step 5 by authenticating S. As a result, the proposed scheme can withstand the denial-of-service attacks.

Performance Analysis

The well-known provable nonmalleable encryption scheme needs 5/3 exponentiations per en/decryption.[9] Since additional hash function operations are needed to transmit the new password in the password change phase, the number of hash operations needed in Hwang and Yeh's scheme and in the proposed scheme are shown as *a/b* in Table 1, where *a* denotes the number of operations needed in the password transmission phase, and *b* denotes the number of operations needed in the password change phase. Moreover, because additional symmetric encryption/decryption operations are also needed in the password change phase of the proposed scheme, the necessary number is represented as above.

The speed of en/decryption with symmetric encryption schemes is faster than that with asymmetric ones. For example, DES is faster than RSA by 1000 times in hardware and 100 times in software.[10] Further, the speed of hash operations is about 1000

Table 1: Number of Operations for Different Types of Computation in Hwang and Yeh's Password Authentication Scheme and the Proposed Authentication Scheme.

Party / Computation type	Hwang-Yeh		The proposed	
	U_i	S	U_i	S
Modular exponential	0(5)	0(3)	0	2
Public key en/decryption	1/0	0/1	0/0	0/0
Symmetric en/decryption	0	0	1/2	1/2
Random number	1	1	1	1
Hash operation	2/4	2/3	2/2	6/7

times faster than RSA operations.[11] Even though the needed number of hash operations in the proposed scheme is more than that needed in Hwang and Yeh's scheme, it is obvious that the scheme presented here is more efficient for the above described reasons. In addition, according to the comparison between Hwang and Yeh's scheme and the new one, it is obvious that the computation load of the user is quite low. That is, the proposed scheme is also suitable in imbalanced networks.

Conclusions

Peyravian and Zunic proposed a simple and efficient password authentication scheme in the year 2000. However, Peyravian and Zunic's scheme is insecure. Hwang and Yeh proposed an improved scheme that uses the server's public key. At the same time, Lee, Li and Hwang presented another improved version of Peyravian and Zunic's scheme. However, there are still security flaws in the scheme proposed by Lee, Li and Hwang. Therefore, this article presents an improvement without employing the server's public key. According to the analyses described above, it is unquestionable that the proposed scheme is secure, practical, and efficient. In addition, the proposed scheme is also suitable in imbalanced networks.

Notes:

[1] B. Clifford Neuman and Theodore Ts'o, "Kerberos: An Authentication Service for Computer Networks," *IEEE Communications Magazine* 32, no. 9 (September 1994): 33-38.

[2] Steven M. Bellovin and Michael Merrit, "Encrypted Key Exchange: Password-Based Protocols Secure against Dictionary Attacks" (paper presented at the 1992 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, California, May 1992), (IEEE Computer Society Press), 72-84.

[3] Chun-Li Lin, Hung-Min Sun, and Tzonelih Hwang, "Three-Party Encrypted Key Exchange: Attacks and a Solution," *ACM Operating Systems Review* 34, no. 4 (2000): 12-20.

[4] Chun-Li Lin, Hung-Min Sun, Michael Steiner, and Tzonelih Hwang, "Three-Party Encrypted Key Exchange without Server Public-Keys," *IEEE Communications Letters* 5, no. 12 (December 2001): 497-499.

[5] Yun Ding and Patrick Horster, "Undetectable On-Line Password Guessing Attacks," *ACM Operating Systems Review* 29, no. 4 (October 1995): 77-86.

[6] Mohammad Peyravian and Nevenko Zunic, "Methods for Protecting Password Transmission," *Computers and Security* 19, no. 5 (2000): 466-469.

[7] Jing-Jang Hwang and Tzu-Chang Yeh, "Improvement on Peyravian-Zunic's Password Authentication Schemes," *IEICE Transactions on Communications* E85-B, no. 4 (April 2002): 823-825.

[8] Cheng-Chi Lee, Li-Hua Li, and Min-Shiang Hwang, "A Remote User Authentication Scheme Using Hash Functions," *ACM SIGOPS Operating Systems Review* 36, no. 4 (October 2002): 23-29.

[9] Ronald Cramer and Victor Shoup, "A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Cipher Attack," in *Proceedings of the 18th Annual International Cryptology Conference on Advances in Cryptology Crypto'98* (Santa Barbara, California, USA, August 1998), published also in *Lecture Notes in Computer Science* 1462 (London, UK: Springer-Verlag, 1998), 13-25.

[10] Bruce Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, Second Edition, (New York: John Wiley & Sons, 1995).

[11] Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*.

YA-FEN CHANG received a B.S. degree in Computer Science and Information Engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2000. She is currently pursuing her Ph.D. degree in Computer Science and Information Engineering from the National Chung Cheng University, Chiayi, Taiwan. Her current research interests include electronic commerce, information security, cryptography, and mobile communications. *Address for correspondence:* Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi 621, Taiwan, R.O.C.; *E-mail:* cyf@cs.ccu.edu.tw.

CHIN-CHEN CHANG received a B.S. degree in Applied Mathematics in 1977 and a M.S. degree in Computer and Decision Sciences in 1979, both from National Tsing Hua University, Hsinchu, Taiwan. He received his Ph.D. in Computer Engineering in 1982 from National Chiao Tung University, Hsinchu, Taiwan. During the academic years of 1980-1983, he was with the Department of Computer Engineering at National Chiao Tung University. From 1983 to 1989, he worked at the Institute of Applied Mathematics, National Chung Hsing University, Taichung, Taiwan. Since August 1989, he has worked as a professor at the Institute of Computer Science and Information Engineering at National Chung Cheng University, Chiayi, Taiwan. Since 2002, he has been a Chair Professor of National Chung Cheng University. His current research interests include database design, computer cryptography, image compression and data structures. Dr. Chang is a fellow of the IEEE, a fellow of the IEE, a research fellow of National Science Council of R.O.C., and a member of the Chinese Language Computer Society, the Chinese Institute of Engineers of the Republic of China, the International Association for Crypto-logic Research, the Computer Society of the Republic of China, and the Phi Tau Phi Honorary Society of the Republic of China. Dr. Chang was the chair and is a honorary chair of the executive committee of the Cryptography and Information Security Association of the Republic of China. *Address for correspondence:* Department of Computer Science and Information Engineering, National Chung Cheng University, 160, San-Hsing, Min-Hsiung, Chiayi 621, Taiwan. *Phone*: 886-5-2720411 ext. 33100; *Fax*: 886-5-2720859; *E-mail*: ccc@cs.ccu.edu.tw.