Password Authentication Schemes with Multi-Servers
能抵抗猜測密碼攻擊之多伺服器通行碼認證機制

Yung-Cheng Lee*, Wen-Chung Kuo† , and Jiin-Ming Hou‡

* : Department of Electrical Engineering,
National Huwei Institute of Technology, Yunlin, 632, Taiwan, R.O.C.
Email: yclee@sunws.nhit.edu.tw

† : Department of Computer Science and Information Engineering,
National Huwei Institute of Technology, Yunlin, 632, Taiwan, R.O.C.

‡: Department of Electrical Engineering,
National Cheng-Kung University, Tainan, 700, Taiwan, 700, R.O.C.

Abstract

Password authentication is an essential approach for users to access a network. Up to now, there are many password authentication schemes proposed. However, many schemes suffer from security risks such as impersonation, replay or guessing attacks, etc. In this paper, we propose two remote password authentication schemes with multi-servers. One scheme is based on geometric approach and the other is based on one-way hash function. Both of the schemes authenticate users without verification tables while the replay and guessing attacks are concealed. Since a one-time pad is used in the login phase, the proposed schemes can efficiently protect a lost smart card.

Keywords: password authentication, smart card protection, guessing attack.

摘要

通行碼認證機制為提供網路使用者登入系統之一重要安全措施。至目前為止，雖有學者提出許多通行碼認證方式，但若干方法可能遭受仿冒、重送與猜測密碼等攻擊。本文以圖形方法與單向函數方法提出二種通行碼認證機制，此二機制之伺服器端均不需驗證表，且能抵抗仿冒、重送與猜測密碼等攻擊。

關鍵詞：通行碼認證，智慧卡保護機制，猜測密碼攻擊。

1. Introduction

In this era of well-developed technologies on computer communications and networks, people frequently access networks or databases through remote login procedures. Unfortunately, due to the wide-open nature of the Internet, there is a security risk associated with all transmitted messages. Thus, the development of a secure authentication approach for remote users is very important.

In a simple remote authentication scheme, the server stores a plain password table in a database. When the user logs in, an identity and a password are entered. The server compares the password with the ones in the password table and access is denied if verification fails. However, because the password is stored in a plain format, there is always the threat of password leakage.

Several schemes use one-way hash functions or cryptographic approaches to encrypt password files in order to avoid the threat of leakage. The hashed or encrypted password files are denoted as verification tables. Although this avoids password leakage, verification tables are still open to the threat of modification. Moreover, they may lead to guessing attacks or replay attacks.

A replay attack is an attacker impersonates a legal user through the reuse of a message obtained in the previous authentication session. Efficient ways to resist replay attacks include using a timestamp or a nonce. Authentication schemes using weak keys such as passwords are vulnerable to guessing attacks. A password guessing attack includes both on-line and off-line attacks. An on-line password guessing attack happens when an attacker attempts to guess the password in an on-line transaction, while an off-line password guessing attack is when an attacker guesses the password and verifies his guess off-line.

In 1981, Lamport [14] proposed a remote authentication scheme with a one-way hash function. The drawback of this scheme, however, is a high hash overhead. Since that time, many proposals have been made to improve efficiency and security [3,4,7,13,22,24,25]. Peyravian and Zunic [19] proposed a method for protecting passwords being transmitted over untrusted networks. Their scheme also used a hash function such as SHA-1 for authentication. But Hwang and Yeh [12] and Lee et al. [17] proved that Peyravian and Zunic's scheme was vulnerable to guessing attacks. However, in the improved system of Lee et al., the probability of guessing the password was not

as low as they declared.

Kwon and Song [13] used a one-time pad and a strong one-way hash function to enhance security and efficiency. Sandirigama [20] also proposed a scheme based on a hash function, but Lin et al. [18] pointed out that their method was vulnerable to replay and denial of service attacks. In 2002, Chen and Ku [5] even proved that the systems of both Sandirigama and Lin et al. were vulnerable to stolen-verifier attacks. An attacker who obtains the password verification table can impersonate a legitimate user in the authentication stage.

Some password authentication schemes are based on geometric approaches. Wu [23] proposed an efficient scheme by using geometric approach on the Euclidean plane. This method allows users to freely change their password and the server does not require a verification table. Unfortunately, Hwang [10] indicated that Wu's scheme was insecure and Chien et al. [3] then proposed a scheme to improve security. Horng and Yang [11] used a certificate based on discrete logarithms to propose a key authentication scheme. Zhan et al. [27] indicated that Horng's scheme was vulnerable to guessing attacks, while Lee et al. [15] pointed out that Zhan's scheme had no non-repudiation feature.

Some authentication schemes allow people to freely choose or update their passwords [4,7-9]. However, people have a tendency to choose passwords that can be easily guessed; successful guessing is therefore feasible if the guess is verifiable and the fail trail is undetectable. Gong et al. [7] proposed an authentication protocol to protect poorly chosen secrets from guessing attacks. Their approach ensured that transmitted data was unpredictable so that off-line verification could not be obtained.

Smart cards, because of their portability and tamper-free features, are convenient and secure devices for use in remote authentication. Though smart cards tend to have deficiencies in computation and memory, they are widely used in modern networks [4,9,26]. Wang and Chang [22] proposed a smart card based password authentication scheme, but security weaknesses were pointed out by Chan [1]. Yang and Shieh [25] used a timestamp with a smart card to propose an ID-based password authentication scheme. The users could freely update their passwords and the server required no verification table. The security was based on the difficulty of factorization. Chan and Cheng [2], however, proved that this system was also insecure, as an attacker could impersonate a valid user. Fan et al. [6] proposed an improved scheme to withstand attacks. Yen and Liao [24] proposed an authentication scheme by using a shared tamperfree cryptographic token to resist replay and weak key attacks; the shared hardware being convenient. Sun [21] proposed a scheme using a hash function instead of modular exponentiation computations. However, Sun's scheme did not provide mutual authentication and users could not freely choose their passwords.

In this paper, we propose two remote password authentication schemes with multi-servers. Both of the schemes authenticate users without verification tables, and the replay and guessing attacks are concealed. Moreover, the proposed schemes can efficiently protect a lost smart card. The remainder of this paper is organized as follows: Two password authentication schemes and their security are reviewed and discussed in Section 2. In Section 3, we proposed two simple password authentication schemes for a multi-server network. Next, we examine the security issues of the proposed schemes in Section 4. Finally, we make some conclusions.

2. Review of the LHL scheme and the CJT scheme

2.1 The LHL scheme

In 2003, Lin et al. [15] proposed a password authentication scheme with multi-servers. Their scheme (hereafter denoted as the *LHL scheme*) is based on geometric approach, users do not need to remember various passwords for different servers. Let $\Gamma = \{S_1, S_2, ..., S_m\}$ denote a set of servers that a user wants to login to, p be a large prime number, g be a primitive root in Galois field GF(p) and PW_i denote the password of user U_i. The LHL scheme is described briefly as follows:

(1) The initialization phase

For each server S_j in Γ, the trusted manager selects a secret key d_j and computes a corresponding public key e_j by

$$e_j = g^{d_j} \pmod{p-1}. \quad (1)$$

(2) The registration phase

The user U_i enters his identity ID_i and password PW_i. The trusted manager computes two points (X_j, Y_j) and (D_{ij}, W_{ij}) with

$$X_j = ID_i^{e_j} \pmod{p},$$

$$Y_j = ID_i^{d_j} \pmod{p}, \quad (2)$$

$$D_{ij} = e_j^{ID_i} \pmod{p},$$

$$W_{ij} = e_j^{PW_i} \pmod{p}, \quad (3)$$

where (X_j, Y_j) is S_j 's secret point and (D_{ij}, W_{ij}) denotes a secret point for U_i to S_j. Then the trusted manager constructs a line $L_{ij} : Y = f_{ij}(X) = aX + b \pmod{p}$ with points (X_j, Y_j) and (D_{ij}, W_{ij}). The trusted manger also selects a public line $LS : Y = g_j(X) = a'X + b' \pmod{p}$, which intercepts line L_{ij} at point (K_{ij}, Q_{ij}).

Let SP_{ij} denote a service period given by server S_j. The trusted manager selects a secret k_{ij} with $(k_{ij}, p-1) = 1$ and calculates

$$r_{ij} = g^{k_{ij}} \pmod{p}. \quad (4)$$

The signature of the service period SP_{ij} is (r_{ij}, s_{ij}), in which s_{ij} is calculated from

$$SP_{ij} = (d_j r_{ij} + k_{ij} s_{ij}) \pmod{p-1}. \quad (5)$$

The trusted manager delivers $\{SP_{ij}, (r_{ij}, s_{ij}), K_{ij}\}$ and line *LS* to the user U_i.

(3) The login phase

If the user wants to login to the server S_j, the user inputs his identity ID_i and password to obtain (D_{ij}, W_{ij}) with Eq.(3). With the public line *LS* and K_{ij}, the user obtains $Q_{ij} = g_j(K_{ij}) \pmod{p}$ and thereby reconstructs line L_{ij} by (D_{ij}, W_{ij}) and (K_{ij}, Q_{ij}). Then the user generates a random number R_{ij}, and calculates A_{ij} and B_{ij} by

$$A_{ij} = g^{R_{ij}} \pmod{p}, \quad (6)$$

$$B_{ij} = e_j^{R_{ij} \times T} \pmod{p}, \quad (7)$$

where *T* denotes the timestamp. The user computes $Z_{ij} = f_{ij}(B_{ij}) \pmod{p}$ and forwards $\{ID_i, (K_{ij}, Q_{ij}), Z_{ij}, A_{ij}, T, SP_{ij}, (r_{ij}, s_{ij})\}$ to the server.

(4) The authentication phase

After receiving the message, the server first checks the validity of ID_i and timestamp *T*. Then the user is authenticated only if the following equation holds:

$$g^{SP_{ij}} = e_j^{r_{ij}} \times r_{ij}^{s_{ij}} \pmod{p}. \quad (8)$$

The server S_j then calculates $B_{ij} = A_{ij}^{d_j T} \pmod{p}$ and obtains line L_{ij} by using points (K_{ij}, Q_{ij}) and (B_{ij}, Z_{ij}). A successful login is obtained only if the secret point (X_j, Y_j) is on the line L_{ij}.

(5) Password update phase

When the user wants to change his password, he first reconstructs line L_{ij} with points (D_{ij}, W_{ij}) and (K_{ij}, Q_{ij}). Through line L_{ij} and public key e_j, the server's secret point (X_j, Y_j) is obtained.

The user selects a new password PW' and obtains (D_{ij}, W_{ij}'), where $W_{ij} = e_j^{PW_i'} \pmod{p}$. A new line L_{ij}' is obtained by points (X_j, Y_j) and (D_{ij}, W_{ij}'). The new intersection point (K_{ij}', Q_{ij}') of line L_{ij}' and line *LS* is obtained. Finally, the user replaces (K_{ij}, Q_{ij}) with (K_{ij}', Q_{ij}') for the next login session.

Security consideration of the LHL scheme

For $D_{ij} = e_j^{ID_i}$ and $W_{ij} = e_j^{PW_i}$ in Eq.(3), if an attacker knows the user's password, he will obtain (D_{ij}, W_{ij}). Through (D_{ij}, W_{ij}) and the public point (K_{ij}, Q_{ij}), the attacker can reconstruct the line L_{ij}. Thereafter, the server's secret point (X_j, Y_j) will be derived by calculating $X_j = ID_i^{e_j} \pmod{p}$ and $Y_j = f_{ij}(X_j) \pmod{p}$.

If the user updates his password for some reason, he will release a new corresponding intersection point (K_{ij}', Q_{ij}'). However, through the points (X_j, Y_j) and (K_{ij}', Q_{ij}'), the attacker can obtain line L_{ij}'. Thus, the user's new secret point (D_{ij}, W_{ij}') also can be obtained by calculating $D_{ij} = e_j^{ID_i}$ and $W_{ij}' = f_{ij}'(D_{ij})$. The attacker can successfully impersonate a legal user by points (D_{ij}, W_{ij}') and (K_{ij}', Q_{ij}'). Therefore, once the password has been compromised,

the system will always be insecure even if the password has been changed.

2.2 The CJT scheme

In 2002, Lee, Hwang and Chien et al. proposed remote authentication systems by using smart cards [4,9,16]. Their methods are similar and have merits such as requiring no verification table, user freely choose/update passwords, resistance to replay attacks and low computation costs. The Chien et al. scheme [4] (hereafter denoted as *CJT scheme*) is described as follows:

(1) The registration phase

Let *x* be the server's secret key and h( ) be a secure one-way hash function. The user submits identity ID_i and password PW_i to the server. The server stores $R_i = h(ID_i \oplus x) \oplus PW_i$ into a smart card and sends it to the user.

(2) The login phase

When the user wants to login to the system, he enters his identity ID_i and password PW_i. The smart card calculates C_i by

$$C_i = h((R_i \oplus PW_i) \oplus T), \quad (9)$$

where *T* is a timestamp. Finally, the user forwards the triplet $\{ID_i, C_i, T\}$ to the server.

(3) The authentication phase

After receiving the message $\{ID_i, C_i, T\}$, the server first checks the validity of ID_i and timestamp *T*. The user is authenticated only if $C_i = h(h(ID_i \oplus x) \oplus T)$.

Security consideration of the CJT scheme

In the CJT scheme, the computation cost is low and the server needs no password verification table. However, because C_i is always fixed for the same timestamp *T*, if an attacker obtains a smart card and knows $\{ID_i, C_i, T\}$ after he intercepts a successful login, he can enter *T* and guess the password by verifying C_i. Therefore, this scheme cannot resist an off-line guessing attack, and an attacker can impersonate the legal user and access the system.

A smart card is a convenient device in today's world, and it protects important security data. Unfortunately, each cardholder may lose his smart card, and if an attacker obtains a card it is possible to successfully guess the password. That is when a smart card is lost, the card holder becomes a security risk

3. The proposed password authentication schemes with multi-servers

We propose two password authentication schemes, which will improve the security. One scheme is based on a geometric approach and the other is based on a one-way hash function. The proposed schemes are compatible with a multi-server network. More importantly, an attacker cannot successfully guess the password even if he obtain the card, so the proposed schemes efficiently protect the owners of lost cards.

3.1 A simple password authentication scheme with a geometric approach

Let *p* be a large prime number, ID_i and PW_i be user U_i's identity and password, respectively. $\Gamma = \{S_1, S_2, ..., S_m\}$ is a set of servers that user U_i wants to login to. There are multi-users, multi-servers, and a trusted manager in the scheme. The trusted manager selects a pair of secret (x_j, y_j) for the server S_j. The proposed scheme is divided into the following four phases: (1) the registration phase, (2) the login phase, (3) the authentication phase, and (4) the password update phase. The scheme is described as follows:

(1) The registration phase

The user U_i sends his identity ID_i and password PW_i to the trusted manager. For user U_i and server $S_j \in \Gamma$, the trusted manager calculates (X_{ij}, Y_{ij}) and (C_{ij}, D_{ij}) as follows:

$$X_{ij} = h(ID_i \oplus x_j),$$
$$Y_{ij} = h(ID_i \oplus y_j), \quad (10)$$
$$C_{ij} = X_{ij} \oplus PW_i,$$
$$D_{ij} = Y_{ij} \oplus PW_i, \quad (11)$$

where $h( )$ denotes a one-way hash function such as MD5 or SHA-1, etc. The trusted manager stores the message $\{ID_i, (C_{ij}, D_{ij})\}$ in a smart card and delivers it to the user U_i.

(2) The login phase

When user U_i wants to login to the server S_j, he inserts his smart card into the card reader and enters password PW_i to obtain (X_{ij}, Y_{ij})

by

$$X_{ij} = C_{ij} \oplus PW_i,$$

$$Y_{ij} = D_{ij} \oplus PW_i. \quad (12)$$

The smart card generates a pair of random numbers (α_i, β_i) as a one-time pad by a pseudo random number generator. Through points (α_i, β_i) and (X_{ij}, Y_{ij}), the line L_{ij}: $y = f_{ij}(x) = a_{ij}x + b_{ij} \pmod p$ can be obtained. Where (a_{ij}, b_{ij}) is obtained by

$$a_{ij} = (Y_{ij} - \beta_i)/(X_{ij} - \alpha_i) \pmod p,$$

$$b_{ij} = ((\beta_i - Y_{ij})/(X_{ij} - \alpha_i)) \times \alpha_i + \beta_i \pmod p. \quad (13)$$

The user U_i calculates $A_{ij} = a_{ij} \oplus X_{ij}$, $B_{ij} = b_{ij} \oplus Y_{ij}$ and $R_i = h(ID_i, a_{ij}, b_{ij}, T_i)$, where T_i denotes the timestamp of the user end. Finally, the user forwards $\{ID_i, A_{ij}, B_{ij}, R_i, T_i\}$ to the server.

(3) The authentication phase

After receiving the message, the server S_j checks the validity of ID_i and verifies whether $|T_i - T_j| \le \Delta T$, where T_j is the current time on the server's computer and ΔT is the expected time interval for transmission delay and clock asynchronicity. Access is denied if verification fails.

Through the secret (x_j, y_j) and user's identity ID_i, the server obtains (X_{ij}, Y_{ij}) by Eq.(10). Thereby (a_{ij}, b_{ij}) is recovered by

$$a_{ij} = A_{ij} \oplus X_{ij},$$

$$b_{ij} = B_{ij} \oplus Y_{ij}. \quad (14)$$

The integrity of the received message can be verified by checking whether $R_i = h(ID_i, a_{ij}, b_{ij}, T_i)$. The process continues only if the integrity is assured.

Next, the line L_{ij} can be reconstructed by (a_{ij}, b_{ij}), and the server checks whether (X_{ij}, Y_{ij}) is located on the line L_{ij}. The point (X_{ij}, Y_{ij}) is on the line L_{ij} if $Y_{ij} = f_{ij}(X_{ij}) = a_{ij}X_{ij} + b_{ij} \pmod p$. A successful login is obtained only if the secret point (X_{ij}, Y_{ij}) is on the line L_{ij}. For illustration, the proposed scheme for user U_1 to server S_1 and S_2 is shown in Fig. 1.

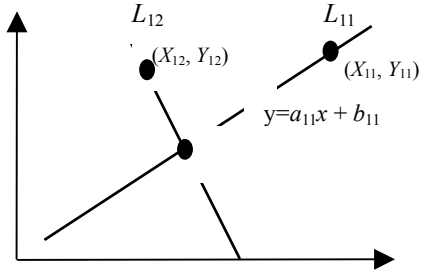

Fig. 1. The concept of the proposed scheme

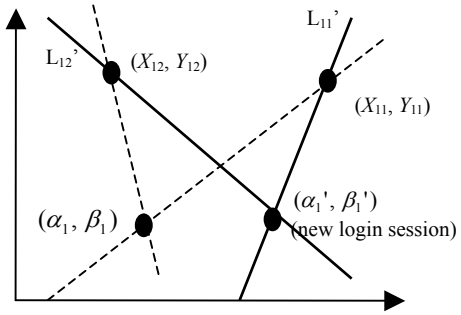

Fig. 2. User's secret point (α_i, β_i) is changed with each login session

Note that the transmitted message is calculated in the smart card and (α_i, β_i) is renewed during each login. The concept of changing (α_i, β_i) with each login session is illustrated in Fig.2.

For mutual authentication, the server calculates $R_j = h(S_j, h(a_{ij}), h(b_{ij}), T_j)$ and forwards $\{S_j, R_j, T_j\}$ to the user, where T_j is the current time of the server's clock. After receiving the response message, the user checks the validity of S_j and T_j. Finally, the server is authenticated if $R_j = h(S_j, h(a_{ij}), h(b_{ij}), T_j)$.

(4) The password update phase

If the user wants to update his password, the process is similar to the CJT scheme [4]. It is not required to inform the server of any password change.

The user enters password PW_i to the smart card to obtain (X_{ij}, Y_{ij}). Next, the user enters a new password PW_i' to calculate and

store new secret information $\{ID_i, (C_{ij}', D_{ij}')\}$ into the card, where $C_{ij}' = X_{ij} \oplus PW_i'$ and $D_{ij}' = Y_{ij} \oplus PW_i'$. From this time on, the user can access the system using the new password PW_i'.

3.2 A simplified password authentication scheme with hash function

The above scheme can be modified to a simplified password authentication scheme. Let all notations be the same as above. The server's secret is x_j. In the registration phase, the steps are similar to those in the previous section. The trusted manager stores $\{ID_i, C_{ij}\}$ in U_i's smart card, where $C_{ij} = X_{ij} \oplus PW_i = h(ID_i \oplus x_j) \oplus PW_i$.

In the login phase, the user enters his password PW_i to obtain X_{ij} by calculating $C_{ij} \oplus PW_i$. The smart card generates a random number α_i as a one-time pad and calculates $A_{ij} = \alpha_i \oplus X_{ij}$. Finally, the user forwards $\{ID_i, A_{ij}, R_i, T_i\}$ to the server, where $R_i = h(ID_i, a_{ij}, T_i)$.

After receiving $\{ID_i, A_{ij}, R_i, T_i\}$, the server checks ID_i and verifies whether T_i is within an expected time interval. Access is denied if verification fails.

Next, the server obtains $X_{ij} = h(ID_i \oplus x_j)$ by using secret x_j and then α_i is recovered by

$$\alpha_i = A_{ij} \oplus X_{ij}$$

Finally, the user is authenticated if $R_i = h(ID_i, \alpha_i, T_i)$.

For mutual authentication, the server calculates $R_j = h(S_j, h(\alpha_i), T_j)$ and forwards $\{S_j, R_j, T_j\}$ to the user. A mutual authentication can be obtained if $\{S_j, R_j, T_j\}$ is verified.

4. Security analysis and discussion

The first proposed scheme has the following merits, and the second scheme is simple with low time and space complexity while security is retained. The discussion on security of the proposed schemes is as follows.

(1) *Resists replay attacks*. Since the transmitted messages $h(ID_i, a_{ij}, b_{ij}, T_i)$ and $h(S_j, h(a_{ij}), h(b_{ij}), T_j)$ include time-stamps, the server or user can detect a replay attack if an attacker re-submits the intercepted message. Note that the system must choose a suitable time interval ΔT so that the replay attack can be avoided while the system clock synchronization and transmission delay problems concealed.

(2) *Resists impersonation attacks*. Since $X_{ij} = C_{ij} \oplus PW_i$ and $Y_{ij} = D_{ij} \oplus PW_i$, the attacker does not know (X_{ij}, Y_{ij}) without a smart card or PW_i. Without (X_{ij}, Y_{ij}), the attacker cannot find the correct (a_{ij}, b_{ij}) to impersonate the valid user. Thus, an impersonation attack can be avoided.

(3) *No password verification table required*. The only secret information required by server S_j is (x_j, y_j), with which the corresponding secret (X_{ij}, Y_{ij}) and in turn (a_{ij}, b_{ij}) can be obtained. Therefore, the server does not require a password verification table for authentication. This simplifies system implementation and key management.

(4) *Provides mutual authentication.* The server's database usually stores important messages and attackers always hope to find ways to retrieve this information. Through the provided mutual authentication approach, the user and server can authenticate each other so that masquerade can be detected.

(5) *Resists on-line password guessing attacks*. In the proposed schemes, the operations at the user end are performed inside the smart card. If an attacker obtains the smart card and tries to guess the password for a login, the server will reject this illegal attempt to access by limiting the number of failed trials. For example, if the server allows an attacker, who has somehow obtained a smart card, to try the password 3 times, the probability for successfully guessing is only $3/2^{|PW|}$, where $|PW|$ denotes the bit length of the password. If the attacker has no smart card, the probability is only $3/2^{2|p|}$ which is much lower than $3/2^{|PW|}$.

(6) *Resists off-line password guessing attacks and protects smart cards*. The off-line guessing attack can be concealed if no verification message is provided. We divided the off-line guessing attack into the following two cases: (a) the attacker has no smart card, and (b) the attacker has stolen or somehow acquired a smart card. For case (a), because the smart card renewed the user's secret point (α_i, β_i) on each login session, thereby (a_{ij}, b_{ij}) also will be changed for each new access. If an attacker tries to guess the exact secret off-line, he will not be successful, because the value (a_{ij}, b_{ij}) is renewed with each session and so he cannot verify his guesses. For case (b), due to the renewed value (a_{ij}, b_{ij}) with each session, if the attacker has obtained the smart card after he has intercepted a successful access and knows (A_{ij}, B_{ij}), he also cannot verify his guess. Therefore, the scheme can resist an off-line guessing password attack and thereby the lost smart card is protected.

(7) *Freedom to update passwords.* Through the smart card, our scheme also allows the user to freely choose or change his password. The user need not inform the trusted manager or servers when updating his password.

5. Conclusions

We have proposed two remote password authentication schemes for multi-servers. One scheme is based on a geometric approach and the other one is based on a secure one-way hash function. Both of the schemes can resist replay and guessing attacks and do not require verification tables. The user can freely choose and update passwords, and mutual authentication is assured. Since a continuously changing secret is used as a one-time pad in the proposed schemes, a guessed password cannot be verified even if the attackers are in possession of a smart card. Therefore, the proposed schemes offer a safe and secure system for the holders of smart cards.

Acknowledgement

This work was supported in part by the National Science Council of the Republic of China under contract number NSC 91-2213-E-150-003.

6. References

[1] C.K. Chan and L.M. Cheng, "Remarks on Wang-Chang's password authentication scheme", Electronics Letters, Vol.37, No.1, 2001, pp.22-23.

[2] C.K. Chan and L.M. Cheng, "Cryptanalysis of a timestamp-based password authentication scheme", Computer & Security, Vol.21, No.1, 2002, pp.74-76.

[3] 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, Is.3, 2001, pp. 287-290.

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

[5] C.M. Chen and W.C. Ku, "Stolen-verifier attack on two new strong password authentication protocols", IEICE Transactions on Communications, Vol.E85-B, No.11, 2002, pp.2519-2521.

[6] L. Fan, J.H. Li, and H.W. Zhu, "An enhancement of timestamp password authentication scheme", Computer & Security, Vol.21, No.7, 2002, pp.665-667.

[7] L.Gong, M.A. Lomas, R.M. Needham, and J.H. Saltzer, "Protecting poorly chosen secrets from guessing attack", IEEE Journal on Selected Area in Communications, Vol.11, No.3, 1993, pp.648-656.

[8] L. Gong, "Optimal authentication protocols resistant to password guessing attacks", Proceedings of the 8th IEEE Computer Security Foundation Workshop, 1995, pp.24-29.

[9] M.S. Hwang, C.C. Lee, and Y.L. Tang, "A simple remote user authentication scheme", Mathematical and Computer Modeling, Vol.36, 2002, pp.103-107.

[10] M.S. Hwang, "Cryptanalysis of a Remote Login Authentication Scheme", Computer Communications, Vol.22, 1999, pp.742-744.

[11] G. Horng and C.S. Yang, "Key authentication scheme for cryptosystems based on discrete logarithm", Computer Communications, Vol.19, 1996, pp.848-850.

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

[13] T. Kwon and J. Song, “Efficient and secure password-based authentication protocols against guessing attack”, Computer Communications, Vol.21, 1998, pp.853-861.

[14] L. Lamport, “Password authentication with insecure communication”, Communications of ACM, Vol. 24, 1981, pp. 770-772.

[15] I.C. Lin, M.S. Hwang, and L.H. Li, “A new remote user authentication scheme for multi-server architecture”, Future Generation Computer Systems, Vol.19, 2003, pp.13-22.

[16] C.C. Lee, M.S. Hwang, and W.P. Yang, “A flexible remote user authentication scheme using smart cards”, ACM Operation Systems Review, Vol.36, No.3, 2002, pp.46-52.

[17] C.C. Lee, L.H. Li, and M.S. Hwang, “A remote user authentication scheme using hash function”, ACM SIGOPS, Operating System Review, Vol.36, Is.4, 2002, pp.23-29.

[18] 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, 2001, pp.2622-2626.

[19] M. Peyravian and N. Zunic, “Methods for protecting password transmission”, Computer & Security, Vol.19, No.5, 2000, pp.466-469.

[20] M. Sandirigama, A. Shimizu and M.T. Noda, “Simple and secure password authentication protocol (SAS)”, IEICE Transactions on Communications, Vol.E83-B, No.6, 2000, pp.1363-1365.

[21] H.M. Sun, “An efficient remote user authentication scheme using smart cards”, IEEE Transactions on Consumer Electronics, Vol.46, No.4, 2000, pp.958-961.

[22] S.J. Wang, and Jin-Fu Chang, “Smart card based secure password authentication scheme”, Computers & Security, Vol.15, No.3, 1996, pp. 231-237.

[23] T.C. Wu, “Remote Login Authentication Scheme Based on a Geometric Approach”, Computer Communications Vol.18, No.12, 1995, pp. 959-963.

[24] S.M. Yen and K.U. Liao, “Shared authentication token secure against replay and weak key attacks”, Information Processing Letters, Vol.62, 1997, pp.77-80.

[25] W.H. Yang, and S.P. Shieh, “Password authentication schemes with smart card”, Computer & Security, Vol.18, No. 8, 1999, pp. 727-733.

[26] T.C. Yeh, H.Y. shen, and J.J. Hwang, “A secure one-time Password authentication scheme using smart card”, IEICE Transactions on Communications, Vol.E85-B, No.11, 2002, pp. 2515-2518.

[27] B. Zhan, Z. Li, Y. Yang, and Z. Hu, “On the security of HY-key authentication scheme”, Computer Communications, Vol.22, 1999, pp.739-741

.