An anonymous mutual authentication system with smart card

Chin-Ling Chen[1] Wei-Chech Lin[2] Zong-Min Guo[3] Yung-Fa Huang[4]
[1,2,3]Department of Computer Science and Information Engineering
Chaoyang University of Technology
[4]Department of Information and Communication Engineering
Chaoyang University of Technology
168 Jifong E. Rd., Wufong Township Taichung County, 41349, Taiwan
[1]clc@mail.cyut.edu.tw; [2]weichech@gmail.com; [3]ckljdstar@gmail.com [4]yfahuang@mail.cyut.edu.tw

Abstract: - User authentication is an important security mechanism for recognizing legal remote users. For this reason, we propose an anonymous mutual authentication scheme for service provider to verify users without verification table. It can resist mostly attacks and ensure the security via improved mutual authentication mechanism, update session key, anonymity, and user can feely change his password. Finally, we make a comparison for security efficiency with other related schemes.

Key-Words: Mutual authentication, RSA, Smart card, Anonymous, Security

1 Introduction

1.1 Related works

Since the remote user tries to access a service on Internet, he or she must make a mutual authentication with the service provider. Therefore, a password based authentication mechanism has adopted widely. In 1981, Lamport [5] proposed a password-based authentication scheme using password tables to authenticate remote users over insecure network. A password table is used to verify a legality of user's identity. But there exist a potential risk as the password table may be stolen or falsified by an attacker, it will influence the system security [2].

To solve the fault of stolen-verifier attack of the Lamport's scheme, Yang and Shieh [8] proposed two remote user authentication schemes without using password tables in 1999. Their scheme used no password table, and maintained the merit of using the mechanism of ID-based such that user can choose and modify their password freely. In 2002, Chan and Cheng [1] presented a forgery attack on Yang and Shieh's timestamp-based password authentication schemes and identified that their schemes are insecure. In 2003, Sun and Yeh [6] pointed out that Chan and Cheng's attack is irrational and has been shown that Yang and Shieh's scheme still suffers from impersonation attack. Afterward, Yang et al. [7] proposed an improvement of Yang and Shieh's timestamp-based and nonce-based password authentication schemes to resist the attack identified by Sun and Yeh in 2005. Due to these schemes perform unilateral authentication schemes (only user's identity can be authenticated), and user has no more information to verify whether the server is authentic or not.

In 2007, Khan [3] showed that Yang et al.'s scheme is still vulnerable to impersonation attack and therefore proposed an improved scheme. In Khan's scheme, he performs the mutual authentication technique to mend the server spoofing attack aim at the security of Yang et al.'s scheme. However, the Khan's scheme is still suffer from ineffective and leaking of user' information. In this paper, The adversarial capabilities allows us to establish a systematic approach for constructing and providing a secure nonce-based password with mutual authentication scheme.

The rest of the paper is organized as follows: In section 2, we present an enhance security mutual authentication scheme . In section 3, we analyze and make a comparison with related works. Finally, we conclude this paper in section 4.

2 Our enhance scheme

To prevent the potential risk described above in former schemes, we propose an enhance security scheme aims to improve the security between remote user and the server.

Our scheme divides into five phases namely system initialization, user registration phase, login phase, authentication phase and uadate password phase. We describe the notations and the steps of each phase as follows.

2.1 Notations

The following notations are used to represent other messages and protocols:

ID_i :	The remote user's identity.
PW_i:	The remote user's password.
U_i:	The i^{th} user
S_j:	The j^{th} server
(n, e)	The public pair key of the server
d	The private key of the server
CID_i :	The dynamic authenticator of the i^{th} user.
k	A secret key
SK :	A session key.
$h(\cdot)$:	A one-way hash function.
▯:	Exclusive-or operation.
N_x:	A random nonce x is generated by x.
r_i:	The i^{th} random number.
||:	The concatenation operation.
$A \overset{?}{=} B$:	Compare whether A equals to B or not
$E_k(M)$	Encryption of a message M using a symmetric key k
M_{upd}	The request of update message

2.2 Initialization phase

In our scheme, a Key Information Center (KIC) is responsible for generating system parameters (such as n, e, d, p, q, $h(\cdot)$, k, and g).

To achieve this, the KIC chooses:

Two randomly and independently large prime numbers p and q.

A RSA modulus:

$$n = p \cdot q \tag{1}$$

A generator g which is the primitive element of $\mathrm{GF}(p)$ and $\mathrm{GF}(q)$.

A collision-resistant hash function $h(\cdot)$ (where $h(\cdot)$ is either SHA-1 or MD5 hash function [7]) which accepts a variant-length input string of bits and produces a fixed-length output string.

The parameters p, q and d, are preserved privately whilst g, n, and the hash function $h(\cdot)$ are publicly known. Once the parameters have been generated, each user U_i shares a secert key k with the server S_j for a login proof.

2.3 User registration phase

Step 1: $U_i \rightarrow S_j$: ID_i and PW_i

The user sends ID_i and PW_i for registering as the legal client

Step 2: $S_j \rightarrow U_i$: smart card

KIC must be generated and published the necessary parameters for every nickname assigned to the user as follows:

$$CID_i = h(ID_i \oplus d) \tag{2}$$

$$S_i = CID_i^{k \cdot d} \bmod n \tag{3}$$

$$T_i = h(PW_i \oplus g) \tag{4}$$

$$h_i = g^{T_i \cdot d} \bmod n \tag{5}$$

$$C_0 = CID_i \oplus h(PW_i) \tag{6}$$

$$R_i = h(T_i \oplus Si) \tag{7}$$

$$K = k \oplus h(PW_i) \tag{8}$$

The KIC uses a nickname CID_i instead the real identity ID_i to protect one's privacy and stores the verifiable information $(n, g, C_0, K, S_i, h_i, R_i, h(\cdot))$ into the smart card.

2.4 Login phase

In the login phase, the user U_i inserts his smart card into the reader and enters his password PW_i.

Step 1: Verify the user is legal or not

The smart card firstly verifies whether the user is legal as follows:

$$T_i^* = h(PW_{\mathrm{i}} \oplus g) \tag{9}$$

$$R_i^* = h(T_i \oplus S_i) \tag{10}$$

$$Check\ R_i^* \overset{?}{=} R_i \tag{11}$$

If the equality holds, the U_i proceeds to acquire the dynamic authenticator CID.

Step 2: $U_i \rightarrow S_j$: CID_i, C_1, V_1

The user U_i computes his or her dynamic authenticator CID_i as follows.

$$CID_i = C_0 \oplus h(PW_i) \tag{12}$$

Afterward, the U_{i} will generate a nonce N_c and computes the following operations.

$$k = K \oplus h(PW_i) \tag{13}$$

$$V_1 = N_C \oplus k \tag{14}$$

$$C_1 = h(CID_i \oplus k \oplus N_C') \tag{15}$$

Then the U_i sends the login request (C_1, V_1, CID_i) to the remote server S_j.

2.5 Authentication phase

Upon receiving the message, the server S_j succeeds in verifying the identity of user U_i by the following equations.

Step 1: $S_j \rightarrow U_i$: C_2, V_2

The S_j receives login message and acquire the nonce

N_C' .

$$N_C' = V_1 \oplus k \tag{16}$$

To verify the correctness of the received login message, the server S_j computes C_1' with the secret key *k*, nonce N_C'

$$C_1' = h(CID_i \oplus k \oplus N_C') \tag{17}$$

And then verify whether the following equality holds or not.

$$\text{Checks } C_1' \overset{?}{=} C_1 \tag{18}$$

If the equality holds, the S_j generates a nonce N_S and computes the response message as follows.

$$C_2 = h(CID_i^{k\cdot d} \bmod n \oplus N_C') \tag{19}$$

$$V_2 = N_S \oplus k \tag{20}$$

Otherwise, rejects the login request.

Step 2: $U_i \rightarrow S_j$: V_3, Y_i

Upon receiving the response message (C_2, V_2), the U_i computes C_2'

$$C_2' = h(S_j \oplus N_C) \tag{21}$$

And verifies its correctness by checking C_2' whether equals to the received C_2 or not

$$\text{Checks } \mathrm{C}_2' \overset{?}{=} \mathrm{C}_2 \tag{22}$$

If the equality holds, the U_i proceeds to acquire the nonce N_S' with his or her secret key *k* and the received V_2.

$$N_S' = V_2 \oplus k \tag{23}$$

To compute the mutual authentication message (V_3, Y_i) , the U_i generates the random number r_i and encrypts with the r_i, N_S' and PW_i into the variable X_i, Y_i and V_3 as follows.

$$X_i = g^{T_\text{i} \cdot r_\text{i}} \bmod n \tag{24}$$

$$Y_i = S_i + h_i^{r_\text{i} \cdot N_S'} \tag{25}$$

$$V_3 = X_i \oplus N_S' \tag{26}$$

Afterward, the U_i will send the message (V_3, Y_i) to the server S_j to request to perform the mutual authentication procedures; otherwise, the U_i will reject the response message.

Step 3: $S_j \rightarrow U_i$: *α, β*

Upon receiving the message (V_3, Y_i), the request of mutual authentication will be confirmed by the S_j. The S_j firstly acquires the verifier X_i by using his or her nonce N_S and the received V_3. To check the validity of the verifier X_i, the S_j also uses the RSA public key *e* to examine the correctness of Y_i as follows.

$$X_i' = V_3 \oplus N_S \tag{27}$$

$$(Y_i)^e \overset{?}{=} CID_i^k \bmod n + X_i'^{N_\text{S}} \tag{28}$$

If the equality holds, the S_j will generate the confirmation message (*α*, *β*) and send it back to the U_i. The computations are shown as below.

$$\alpha = h(CID_i \,\|\, Y_i \,\|\, X_i' \,\|\, N_C' \,\|\, N_S \,\|\, k) \tag{29}$$

$$\beta = \alpha \oplus N_S \tag{30}$$

Step 4: $U_i \rightarrow S_j$: M_{upd}

After receiving the confirmation message (α, β), the U_i will check its correctness.

$$\beta \oplus N_S' \overset{?}{=} h(CID_i \,\|\, Y_i \,\|\, X_i \,\|\, N_C \,\|\, N_S' \,\|\, k) \tag{31}$$

If the equality holds, the U_i will continuously regenerate the session key *SK* and send the updating message M_{upd} back to the S_j. The user computes the session key *SK* as below:

$$SK = g^{h(Nc\|Ns'\|\alpha)} \tag{32}$$

step5:$U_i \longleftrightarrow S_j$:

Upon receiving the updating message M_{upd}, the S_j computes the newly session key *SK* and executes the procedure of replacing the session key *SK* .

$$SK = g^{h(N_\text{C}'\| N_\text{S} \| \alpha)} \tag{33}$$

Thus, both the requirements of mutual authentication and session key *SK* agreement can therefore be achieved after the authentication phase.

2.6 Update password phase

Step1: U_i: Update secret factors

The user inputs a new password $PW_{i_{NEW}}$ and then the smart card will compute the new secret parameters ($C_{0_{new}}$, $T_{i\,new}$, $h_{i_{NEW}}$, $R_{i\,new}$, K_{new})

$$T_{i_{NEW}} = h(PW_{i_{NEW}} \oplus g) \tag{34}$$

$$C_{\mathbf{0}\,NEW} = CID_i \oplus PW_{i_{NEW}} \tag{35}$$

$$h_{i_{NEW}} = g^{T_i \cdot d} \bmod n \tag{36}$$

$$R_{i\,NEW} = h(T_i \oplus S_i) \tag{37}$$

$$K_{NEW} = k \oplus h(PW_{i_{NEW}}) \tag{38}$$

And then stores the new parameters into smart card.

3 Analysis and Discussions

3.1 Security analysis

3.1.1 Replay attack issue

In authentication phase, the adversary may play a replay attack by resending the authenticated messages and could be succeeded between the communication parties is unchangeable. In our scheme, all nonce (i.e., N_C and N_S) are variable and would be verified by another party during the communication. The verification equations are shown as following eqautions:

$$C_2' \overset{?}{=} h(S_i \oplus N_C)$$

$$(Y_i)^e \overset{?}{=} CID_i^k \bmod n + X_i{}'^{N_S'}$$

It is clearly that our proposed scheme can resist the replay attack.

3.1.2 Forgery attack issue

The transaction messages of our proposed scheme are protected by cryptographic mechanism. If an adversary expects to forge a legal message (for example: V_1, V_2), it is necessary to get the secret key *k*. Since the secret key *k* has only shared between the communication parties. Thus, the attackers cannot obtain the secret key *k*. On other hand, some message (for example: C_1, C_2, Y_i and α) are protected under the collision-resistant hash function $h(\cdot)$. Therefore, it is computing infeasible to the adversary to extract the secret key *k* directly.

3.1.3 Insider attack issue

If the insider attacker stole (n, e, d, k) from the database, impersonated the legal server and derived user's real identity or breached secure authentication scheme. However, the secret key *k* is held by asminstrator and nerver transmit to other people. In authentication phase, it needs input the authority delegation secret key *k*, as shown in below equations:

$$N_C' = V_1 \oplus k$$

$$C_2 = h(CID_i^{k \cdot d} \bmod n \oplus N_C')$$

The attacker cannot pass the user authentication as following equation:

$$C_2' \overset{?}{=} h(S_i \oplus N_C)$$

Consequently, the insider wants to carry on illegal access is impossible.

3.1.4 Forward secrecy issue

The attacker might intercept the message argument (C_1, V_1, CID_i). Because the messages are ciphertext, the attacker cannot decrypt and derive user's password PW_i and secret key *k* via the collision-resistant hash function $h(\cdot)$, the protected messages are shown in below

$$CID_i = C_0 \oplus h(PW_i)$$

$$C_1 = h(CID_i \oplus k \oplus N_C')$$

Therefore, the attacker cannot intercept information form communication messages and impersonate a legal user.

3.1.5 Parallel session attack issue

In login and authenticatin pahse, the attackers intercepted the verifiers, he/she cannot derive or modify any messages. In our scheme, user sends a login request message (C_1, V_1, CID_i) to server, and the message (C_2, V_2) was sent from server to user. The related messages are shown as follows:

$$CID_i = h(ID_i \oplus d)$$

$$C_2 = h(CID_i^{k \cdot d} \bmod n \oplus N_C')$$

A result of the attacker does not hold password PW_i and secret nunber *d*, thus he/she cannot intercept the message and modify it. Therefore, he/she can not tamper a legal verifer CID_i and C_2 as following equations:

$$C_1' = h(CID_i \oplus k \oplus N_C') \neq C_1$$

$$C_2' = h(S_i \oplus N_C) \neq C_2$$

The user and server reject the authentication requests. Therefore, our schem can resist parallel session attacks.

3.2 Anonymity issue

In our proposed scheme, the U_i has maintained the property of anonymity aim at his or her identity even if the adversary could intercept the communication message. Without any knowledge of the private key *d* or the U_i's personal password PW_i, it is unable to the adversary to know or to gain the real identity refers to the intercepted C_0 or CID_i as following equations:

$$CID_i = h(ID_i \oplus d)$$

$$C_0 = CID_i \oplus h(PW_i)$$

Therefore, the anonymity property in our scheme can easily be achieved.

3.3 Mutual authentication issue

In order to provide the proof to each communication parties, the mutual authentication issue is also discussed in our proposed scheme. At the server side, the S_j can confirm the legality of the U_i by verifying the following equation.

$$C_1' \overset{?}{=} h(CID_i \oplus k \oplus N_C')$$

Also the U_i can confirm the legality of the S_j by verifying the following equation.

$$C_2' \overset{?}{=} h(S_i \oplus N_C)$$

Afterward, the S_j performs mutual authentication message by checking the correctness of X_i' and Y_i as following equation:

$$X_i' = V_3 \oplus N_S$$

$$(Y_i)^e \overset{?}{=} CID_i^{\,k} \bmod n + X_i^{N_S}$$

Continuously, the session key agreement procedure has been started. If the above equation holds, the S_j performs the computing of the verifiers α and β as follows equations:

$$\alpha = h(CID_i \,\|\, Y_i \,\|\, X_i' \,\|\, N_C' \,\|\, N_S \,\|\, k)$$

$$\beta = \alpha \oplus N_S$$

At next, the U_i can also verify the validity of α and β.

$$\beta \oplus N_S' \overset{?}{=} h(CID_i \,\|\, Y_i \,\|\, X_i \,\|\, N_C \,\|\, N_S' \,\|\, k)$$

Finally, both of the U_i and S_j compute the newly session key *SK* and replace the old session key *SK* as following equations:

$$SK = g^{h(N_\mathrm{C} \,\|\, N_\mathrm{S}' \,\|\, k)} = g^{h(N_\mathrm{C}' \,\|\, N_\mathrm{S} \,\|\, k)}$$

Therefore, it is clearly that our scheme can complete the purpose of mutual authentication by the verifiable proofs.

3.4 Two-factor security issue

For our improved scheme, the parameters $(n, g, C_0, K, S_i, h_i, R_i, h(\cdot))$ within the smart card are hard to derive if the attacker has obtained the user's password instead of smart card. The attacker may also intercept the user's previous login request messages (C_1, V_1, CID_i), it is infeasible to derive nonce N_C and ID_i fromV_1 and CID_i which are based on the security of one-way hash function. Similarly, N_S and r_i are hard to extract from V_2 and Y_i. On the other hand, if the attacker steals the user's smart card and extracts the parameter values (n, g, C_0, k, S_i, h_i) stored in the smart card with some ways, he/she still cannot obtain PW_i directly. Thus, our scheme can provide two-factor security.

3.5 Freely change password issue

He/she can freely input new password and update the related secret parameters (C_0, T_i, h_i, R_i, K) about password PW_i in smart card as following equations:

$$T_{i_{NEW}} = h(PW_{i_{NEW}} \oplus g)$$

$$C_{\mathbf{0}_{NEW}} = CID_i \oplus PW_{i_{NEW}}$$

$$h_{i_{NEW}} = g^{T_i \cdot d} \bmod n$$

$$R_{i_{NEW}} = h(T_i \oplus S_i)$$

$$K_{NEW} = k \oplus h(PW_{i_{NEW}})$$

Therefore, we support a dynamic change password scheme.

3.6 Security comparisons

Comparison of the proposed scheme and previously schemes is depicted in Table 1, from which it can be seen that the Yang et al., Kim et al. and Khan's schemes are all neither withstand the secure attack and leak of password nor achieve mutual authentication and user anonymity. As well as the proposed scheme constructs the session key implicitly on performing user identification, requiring no extra overhead. The update password phase insures the password is secure and does not need the server to update password simultaneously. In addition, the proposed scheme further provides against parallel session key confirmation between each party.

3.7 Computation cost evaluation

Because exclusion-OR operation requires very few computation, and the SHA operation can be bounded in a constant time. These computational costs usually be neglected consisdering.We make a comparison of the computation cost with other related schemes in Table 2. In spite od the cost of Yang's scheme are minimun. However their scheme cannot satisfy the complete security requirements. At user side, The cost of our scheme (2 T_{mod} +2 T_{em}+3 T_{eo}+5 T_h) is almost equal to Kim's scheme (2 T_{mod} +2 T_{em}+3 T_{eo}). At server side, our scheme is higher, but we supported a password change phase, achieved the mutual authentication and complete secure requirements. In general, our scheme is superior to previous schemes.

4 Conclusion

In this paper, we have proposed an effective scheme in which supported password change, mutual authentication, prevented a serial of attacks. Besides, we make a comparison with previous schemes in table 1 and 2. According to the serial of comparisons, it is clearly that our scheme can resist mostly attacks, support the securities, and it is available. In the future, we hope that the nonce-based mutual authentication technique can be widely adopted and expanded in smart card-based or mobile device-based schemes.

Table 1.The comparisons of our proposed scheme and previous schemes.

	Yang et al. [6]	Kim et al.[4]	Khan[3]	Our scheme
Against replay attack	N	N	N	Y
Against forgery attack	N	N	N	Y
Against inside attack	N	N	N	Y
Against parallel session attack	N	N	N	Y
Anonymity	N	N	N	Y
Forward secrecy	N	N	N	Y
Mutual authentication	N	N	N	Y
Against the leak of password	N	N	N	Y
No time-synchronization problem	N	N	N	Y
Early detection	N	N	N	Y
Freely change the password	N	N	N	Y

Table 2.The comparisons of the computation cost

Scheme	Yang et al. [12]	Kim et al.[6]	Khan[5]	Our scheme
User	$1T_{mod}+2T_{em}$ $+2\ T_{eo}$	$2\ T_{mod}+2\ T_{em}$ $+3\ T_{eo}$	$2\ T_{mod}+2\ T_{em}$ $+2\ T_{eo}+1\ T_h$	$2\ T_{mod}+2\ T_{em}+3\ T_{eo}+5\ T_h$
Server	$3\ T_{eo}$	$1\ T_{mod}+1\ T_{em}$ $+3\ T_{eo}$	$1\ T_{mod}+1\ T_{em}$ $+4\ T_{eo}+2\ T_h$	$1\ T_{mod}+1\ T_{em}+5\ T_{eo}+4\ T_h$

Note: T_{mod}: time complexity of the modular operation, T_{em}: time complexity of the exponent multiplication, T_{eo}: time complexity of the exponent operation, T_h: time complexity of the one-way hash function

References:

[1] Chan, C. K. and Cheng, L. M., Cryptanalysis of a Timestamp-Based Password Authentication Scheme, *Computers & Security*, Vol. 21, No.1, pp. 74f-76, 2002.

[2]Hwang, M.S. and Li, L. H., A New Remote User Authentication Scheme using Smart Cards, *IEEE Transactions on Consumer Electronics*, Vol. 46, No. 1, pp. 28-30, 2000.

[3]Khan, M. K., Cryptanalysis and Security Enhancement of Two Password Authentication Schemes with Smart Cards, *IEEE International Multitopic Conference (INMIC'07)*, pp. 1-4, 2007.

[4]Kim, K. W., Jeon, J. C. and Yoo, K. Y., ”An improvement on Yang et al.'s password authentication schemes,” Applied Mathematics and Computation, Vol. 170, No.1, pp. 207-215, 2005.

[5]Lamport, L., Password Authentication with Insecure Communication, *Communications of the ACM*, Vol. 24, No. 11, pp. 770-772, 1981.

[6]Sun, H. M. and Yeh, H. T., Further Cryptanalysis of a Password Authentication Scheme with Smart Cards, *IEICE Transactions on Communications*, Vol. 86B, No. 4, pp. 1412-1415, 2003.

[7]Yang, C. C., Wang, R. C. and Chang, T. Y., An Improvement of the Yang-Shieh Password Authentication Schemes, *Applied Mathematics and Computation*, Vol. 162, No. 3, pp. 1391-1396, 2005.

[8]Yang, W. H. and Shieh, S. P., Password Authentication Schemes with Smart Cards, *Computers & Security*, Vol. 18, No. 8, pp. 727-733, 1999.