A nonce-based mutual authentication system with smart card

Chin-Ling Chen¹ Wei-Chech Lin² Zong-Min Guo³ Yung-Fa Huang^{4,} Neng-Chung Wang⁵ ^{1,2,3}Department of Computer Science and Information Engineering Chaoyang University of Technology ⁴Department of Information and Communication Engineering Chaoyang University of Technology ^{1,2,3,4}168 Jifong E. Rd., Wufong Township Taichung County, 41349, Taiwan (R.O.C.) ⁵Department of Computer Science and Information Engineering National United University of Technology ⁵1, Lienda, Miaoli, 36003, Taiwan (R.O.C) ¹clc@mail.cyut.edu.tw; ²weichech@gmail.com; ³ckljdstar@gmail.com ⁴yfahuang@mail.cyut.edu.tw; ⁵ncwang@nuu.edu.tw

Abstract: - User authentication is an important security mechanism for recognizing legal remote users. We propose an available and secure authentication scheme for service provider to verify users without using verification table. It can resist most of the attacks by improving nonce-based mutual authentication mechanism, and ensure the security by dynamic session key. User may change his password freely. Our scheme compared with other related schemes for security efficiency.

Key-Words: - mutual authentication, RSA, smart card

1 Introduction

1.1 Related works

To access a service over Internet, a remote user should make a mutual authentication with the service provider. Currently, a password based authentication mechanism is widely used. In 1981, Lamport [9] 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 exists a potential risk as the password table may be stolen or falsified by an attacker; it violates the system security.

To solve the fault of stolen-verifier attack of the Lamport's scheme, Yang and Shieh [16] proposed nonce-based and timestampe-based 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 [14] 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. [15] 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. These schemes provide one-way authentication schemes for certificate (only user's identity). However, the server's identification is not authenticated.

In 2007, Khan [7] showed that Yang et al.'s scheme is still vulnerable to impersonation attack and therefore proposed an improved scheme. In Khan's scheme, the mutual authentication technique is used 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.

1.2 Our results

In this paper, we proposed a new set of security requirements for nonce-based password mutual authentication. The new set simplified previously proposed set, it removed and ambiguities in some part of requirmentset, it also facilitates cryptanalysis in a better way with an adversarial model. The new requirement set is also association with an adversarial model. The separation of requirement set and adversarial capabilities allows us to establish a systematic approach for constructing and providing a secure nonce-based password with mutual authentication scheme.

1.3 Requirements

In order that to hide user information and hold integrated identity authentication scheme, we considers the proposed scheme should resist the following the attacks.

1.Replay attack: the attacker intercepts the message, and transmits the old message again for masquerade a legal user or server.

2.Inside attack: If an attacker is a system inside member, he can get some secrets and pretend a legal server.

3.Know-key attack [2] : if the session key leaked out and the attacker obtains it, he or she can use the session key to decrypt the ciphertext.

4.DoS attack: A attacker delivers a fake message to a server continuously and a server's response time was delayed for a long time, the service was interrupted.

5.Anonymity [5] [11] : The user does not leak out his real dientity during the transaction.

6.Forward secrecy: The attacker permeates through the interception messages to obtain any earlier information.

7.Mutual authentication [4] : The user or server can verify identity mutually.

8.Dynamic session key[10]: It ensure the key is secure and dynamic update.

9. Time-synchronization: The remote user and server's time should be identical.

10.Freely change password: If a legal user considers his password is risk, he can freely change password at any time.

1.4 Preliminaries

In order to satisfy the requirements, we intend to integrate the cryptographic mechanisms to implement a nounce-based mutual authentication system. In this section, we will describe the one-way hash function.

The one-way hash function requires a variable-length input string and converts it to a fixed-length input string. It works in one-way direction. It is easy to compute a hash value from an original text, but it is hard to generate a pre-image that hashes to a particular value. For example, giving an input x, it is easy to compute the output y = h(x) through the function and from an output y, it is computationally

infeasible to derive the input x that satisfies h(x) = y. If giving an input x, it is difficult to find another input x' such that the two inputs have the same output y such that h(x) = h(x') = y.

SHA (Secure Hash Algorithm) [13] is a well-know one-way hash function and its output is a 160 bits hash value. Besides the properties of the one-way hash function and collision-resistant, it also provides the benefits of low computational overhead and easy software implementation [12].

1.5 Paper organization

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. The scenarios of our proposed improvement scheme are illustrated in Fig. 1.

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>i</i> th user
S_j :	The <i>j</i> 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 collision-resistant 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.
$A \stackrel{?}{=} B$:	Compare whether A equals to B or not

$E_k(M)$	Encryption of a message M using a symmetric key k	numbers <i>p</i> and <i>q</i> . A RSA modulus [3]:	
M_{upd}	The request of update message	$n = p \cdot q$	(1)

A generator g which is the primitive element of GF(p) and GF(q).

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 A collision-resistant hash function $h(\cdot)$ (where $h(\cdot)$ is either SHA-1 or MD5 hash function [13]) 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



Fig. 1. Our improved nonce-based password authentication scheme

while g, n, and the hash function $h(\cdot)$ are publicly known. Once the parameters have been generated, each user U_i shares a secent 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: $S_i \rightarrow U_i$: smart card

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

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

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

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

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

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

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

$$A = k^e \tag{8}$$

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

$$T_i^* = h(PW_i \oplus g) \tag{9}$$

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

Check
$$R_i^* \stackrel{?}{=} R_i$$
 (11)

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

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

The user U_i computes his/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.

$$V_1 = N_C \oplus A \tag{13}$$

$$C_1 = h(CID_i \oplus A \oplus N_C') \tag{14}$$

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

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' .

$$B = k^e \tag{15}$$

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

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

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

And then verify whether the following equality holds or not.

$$\operatorname{Checks} C_1 \stackrel{?}{=} C_1 \tag{18}$$

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

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

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

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

Checks
$$C_2 = C_2$$
 (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 .

$$N_{S}' = V_{2} \oplus A \tag{23}$$

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.

$$X_i = g^{T_i \cdot r_i} \mod n \tag{24}$$

$$Y_{i} = S_{i} + h_{i}^{r_{i} \cdot N_{S}'}$$
(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: $S_j \rightarrow U_i: \alpha, \beta$

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.

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

$$(Y_i)^e \stackrel{?}{=} CID_i^k \mod n + X_i^{N_s}$$
 (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.

$$\alpha = h(CID_i || Y_i || X_i' || N_C' || N_S || B)$$
(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.

$$\beta \oplus N_{s}' \stackrel{?}{=} h(CID_{i} || Y_{i} || X_{i} || N_{C} || N_{s}' || A) \quad (31)$$

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:

$$SK = g^{h(NC||NS'||\alpha)}$$
(32)

step5: $U_i \leftarrow \rightarrow S_j$:

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 with

а.

User U_i Update key phase Enter new password $PW_{i_{NEW}}$ $h_{i_{NEW}} = g^{PW_{i_{NEW}} \cdot d} \mod n$ $C_{0_{NEW}} = CID_i \oplus PW_{i_{NEW}}$ $T_{i_{NEW}} = h(PW_{i_{NEW}} \oplus g)$ $R_{i_{NEW}} = h(T_i \oplus S_i)$



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 $(h_{i_{new}}, C_{0_{new}}, T_{i_{new}}, R_{i_{new}})$

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

$$h_{i_{NEW}} = g^{T_{i_{NEW}} \cdot d} \mod n \tag{35}$$

$$C_{\mathbf{0}_{NEW}} = CID_{i} \oplus PW_{i_{NEW}} \tag{36}$$

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

And then stores the new parameters into smart card.

3 Analysis and Discussions

3.1 Security analysis

3.1.1 DoS attack issue

The attacker resends the previous login messages (C_1, V_1, CID_i) and expects to pass the server's verification. Unfortunately, it will not succeed as the resend message can be detected by the server S_j . Because of the C_1 is made by the secret key k and nonce N_C ' as shown in below equations:

$$C_{1}' = h(CID_{i} \oplus B \oplus N_{C}')$$

Checks $C_{1}' \stackrel{?}{=} C_{1}$



And the session key *SK* will be updated when an authentication session is done. The equation is as follows:

$$SK = g^{h(N_{\rm C} \parallel N_{\rm S}' \parallel \alpha)}$$

Thus, it is infeasible to the adversary to palsy our scheme by resending the previous login messages sunceasingly. Our proposed scheme can resist the DoS attack.

3.1.2 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., $N_{\rm C}$ and $N_{\rm S}$) are variable and would be verified by another party during the communication. The verification equations are shown as following eqautions:

$$C_{2}' \stackrel{?}{=} h(S_{i} \oplus N_{C})$$

$$(Y_{i})^{e} \stackrel{?}{=} CID_{i}^{k} \mod n + X_{i}'^{N_{S}'}$$

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

3.1.3 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 **a**) 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.

3.1.4 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:

$$N_C' = V_1 \oplus B$$

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

The attacker cannot pass the user authentication as following equation:

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

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

3.1.5 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

$$CID_i = C_0 \oplus h(PW_i)$$
$$C_1 = h(CID_i \oplus A \oplus N_C)$$

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

3.1.6 Parallel session attack issue

In login and authenticatin pahse, the attackers intercepted the verifiers, they 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:

$$CID_{i} = h(ID_{i} \oplus d)$$
$$C_{2} = h(CID_{i}^{k \cdot d} \mod n \oplus N_{C}')$$

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

$$C_{1}' = h(CID_{i} \oplus B \oplus N_{C}') \neq C_{1}$$
$$C_{2}' = h(S_{i} \oplus N_{C}) \neq C_{2}$$

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

3.1.7 know-key attack issue

Our scheme uses the ephemeral nonces N_C and N_S in the authentication pahse. Nonces are randomly and independent in each pahse.

Moreover, the session key *SK* is established by a smart card and server in each session as following equation:

$$SK = g^{h(NC||NS'||\alpha)} = g^{h(NC'||NS||\alpha)}$$

Because the session key is also independent. Therefore, the knowledge of previous session key can not derive a new session key. As a result, the know-key attack does not work in our proposed scheme.

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 dor 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.

$$C_1' \stackrel{?}{=} h(CID_i \oplus B \oplus N_C')$$

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

$$C_2' \stackrel{?}{_} h(S_i \oplus N_C)$$

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

$$X_{i} '=V_{3} \oplus N_{S}$$

$$(Y_{i})^{e} = (S_{i} + h_{i}^{T_{i} \cdot N_{S}} \mod n)^{e}$$

$$= (CID_{i}^{k \cdot d} \mod n + g^{T_{i} \cdot d \cdot r_{i} \cdot N_{S}} \mod n)^{e}$$

$$= CID_{i}^{k \cdot e \cdot d} \mod n + g^{T_{i} \cdot e \cdot d \cdot r_{i} \cdot N_{S}} \mod n$$

$$= CID_{i}^{k} \mod n + g^{T_{i} \cdot r_{i} \cdot N_{S}} \mod n$$

$$= CID_{i}^{k} \mod n + X_{i}^{N_{S}}$$

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:

$$\alpha = h \left(CID_i \parallel Y_i \parallel X_i' \parallel N_C' \parallel N_S \parallel B \right)$$

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

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

$$\beta \oplus N_{S'} \stackrel{?}{=} h(CID_{i} \parallel Y_{i} \parallel X_{i} \parallel N_{C} \parallel N_{S'} \parallel A)$$

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:

$$SK = g^{h(N_{c} ||N_{s}'||\alpha)} = g^{h(N_{c}'||N_{s}||\alpha)}$$

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

3.4 Two-factor security issue

If both of the user's smart card and his password were stolen, then there is no way to prevent the attacker from masquerading as the user. So the best policy we can do is to guarantee the security of the scheme when either the user's smart card or his password is secure, but not both. This security property is called two-factor security. For our improved scheme, the parameters (n, g, C_0, A, S_i, h_i) 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 from V_1 and CID_i which are based on the security of collision-resistant 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 $(n, g, C_0, A, S_i, h_i, R_i, h(\cdot))$ stored in the smart card with some ways, he or she still cannot obtain PW_i directly. Thus, our scheme can provide two-factor security.

3.5 Early detection issue

When the user inputs wrong password, the smart card detects the error right now. If the attacker gets the smart card and inputs a forgery password PW_i^{2} . The samrt card verifies the legality with the password as following equations:

$$T_{i}' = h(PW_{i}' \oplus g)$$

$$R_{i}' = h(T_{i}' \oplus S_{i})$$

$$R_{i}' \neq h(h(PW_{i} \oplus g) \oplus S_{i})$$

Therefore, it is clearky that our scheme can early detecation the attacker or wrong password.

3.6 Freely change password issue

User can change old password via the smart card freely. If user thinks his/her password has risk or suffer from attack. He/she can freely input new password and update the related secret parameters (h_i, C_0, T_i, R_i) about password PW_i in smart card as following equations:

$$T_{i_{NEW}} = h(PW_{i_{NEW}} \oplus g)$$
$$h_{i_{NEW}} = g^{T_{i_{NEW}} \cdot d} \mod n$$
$$C_{0 NEW} = CID_{i} \oplus PW_{i_{NEW}}$$
$$R_{i_{NEW}} = h(T_{i} \oplus S_{i})$$

Therefore, we support a dynamic change password scheme.

3.7 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 reflection attack, DoS 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.

3.8 Communication cost evaluation

In this section, we compare the communication cost of our scheme to previous schemes in table 2. We use one-way hash function, exclusive-or operation, modular operation and nonce-based to assure secure requirements. Although the cost of the round time in Yang et al.'s and the Kim's is lower than other, but they cannot achieve mutual authentication. Our cost of round time is higher than others, but our scheme assures the mutual authentication requirement. Our scheme is more secure than others.

3.9 Computation cost evaluation

We compare the computation cost of our scheme to previous schemes in table 3. 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 consistering. Thus, we make a comparison of the computation cost with other related schemes in Table 3. 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 $(2 T_{mod} + 1 T_{em} + 5 T_{eo} + 4 T_h)$ 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 and achieved the mutual authentication and complete secure requirements. In general, our scheme is superior to previous schemes.

	Yang et al. [14]	Kim et al.[8]	Khan[7]	Our scheme
Against replay attack	Ν	Ν	Ν	Y
Against DoS attack	Ν	Ν	Ν	Y
Against inside attack	N	Ν	Ν	Y
Against parallel session attack	N	Ν	Ν	Y
Against know-key attack	Ν	Ν	Ν	Y
Anonymity	Ν	Ν	Ν	Y
Forward secrecy	Ν	Ν	Ν	Y
Mutual authentication	Ν	Ν	Ν	Y
Against the leak of password	N	Ν	Ν	Y
No time-synchronization problem	Ν	Ν	Ν	Y
Early detection	N	N	N	Y
Freely change the password	N	N	N	Y

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

Scheme	Yang et al. [14]	Kim et al.[8]	Khan[7]	Our scheme
Cost of round time	3	3	4	5
Cost of bit length	1 M +5 C	2 M +1 H +5 C	2 M +2 H +5 C	2 M +5 H +2 N

Table 2. The comparisons of the communication cost

Note: |M|:the bit length of modular

|H|:the bit length of SHA hash function |N|:the bit length of the nonce

|C|: the bit length of the constant

Table 3.The comparisons of the computation cost

Scheme	Yang et al. [14]	Kim et al.[8]	Khan[7]	Our scheme
User	$1T_{mod}+2T_{em}$ +2 T _{eo}	$\begin{array}{c} 2 \operatorname{T}_{mod} + 2 \operatorname{T}_{em} \\ + 3 \operatorname{T}_{eo} \end{array}$	$\begin{array}{c} 2 \operatorname{T}_{mod} + 2 \operatorname{T}_{em} \\ + 2 \operatorname{T}_{eo} + 1 \operatorname{T}_{h} \end{array}$	$2 T_{mod} + 2 T_{em} + 3 T_{eo} + 7 T_h$
Server	3 T _{eo}	$\begin{array}{c} 1 \operatorname{T}_{mod} + 1 \operatorname{T}_{em} \\ + 3 \operatorname{T}_{eo} \end{array}$	$1 \operatorname{T}_{mod} + 1 \operatorname{T}_{em} \\ + 4 \operatorname{T}_{eo} + 2 \operatorname{T}_{h}$	$2 T_{mod} + 1 T_{em} + 5T_{eo} + 4 T_h$

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

4 Conclusion

In this paper, we have proposed an effective scheme in which supported password change, mutual authentication, prevented a serial of attacks. These attacks include replay attack, DoS attack, insider attack, forgery attack, forward secrecy and parallel session attack etc.

On the other hand, we also keep secure features as listed below:

- (1) Anonymity.
- (2) Mutual authentication.
- (3) Two-factor security.
- (4) Key agreement.
- (6) Freely change password.

Besides, we make a comparison with previous schemes in table 1, 2 and 3. 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.

References

- 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]Chang C. C., Lee, C. Y., Chiu, Y. C., Enhanced authentication scheme with anonymity for

roaming service in global mobility networks, Computer Communications, Vol. 32, No. 4, pp. 611-618, 2009.

- [3] Lupu C., Firtat B. and Enoiu C., Cryptography Methods Using the RSA Algorithms, *WSEAS Transactions on Communications*, Vol. 4, No. 4, pp.153-156, 2005
- [4]Emmanuel Bresson, Olivier Chevassut, Abdelilah Essiari, David Pointcheval, Mutual authentication and group key agreement for low-power mobile devices, *Computer Communications*, Vol. 27, No. 17, pp. 1730-1737, 2004.
- [5]Fung K., Chan T. K., Liu, J. K. and Wong, D. S., Threshold Anonymous Credential System, *WSEAS Transactions on Communications*, Vol. 3, No. 2, pp. 839-842, 2004.
- [6]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.
- [7]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.
- [8]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.
- [9]Lamport, L., Password Authentication with Insecure Communication, *Communications of the ACM*, Vol. 24, No. 11, pp. 770-772, 1981.
- [10]Lee, D. G and Lee, I. Y., A Study on encrypted

key exchange using password, *WSEAS Transactions on Communications*, Vol. 3, No. 2, pp. 503-510, 2004.

- [11] Lewis, P. H., Computer Jokes and Threats Ignite Debate on Anonymity, *The New York Times*, pp. 155, 1994
- [12]Merkle, R. C., A fast software one-way hash funcation, *Journal of Cryptology*, Vol. 3, No. 1, pp. 43-58, 1990
- [13]Sarkar, P., Domain extender for collision resistant hash functions: Improving upon Merkle–Damgard iteration, *Discrete Applied Mathematics*, 2008.
- [14]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.
- [15]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.
- [16]Yang,W. H. and Shieh, S. P., Password Authentication Schemes with Smart Cards, *Computers & Security*, Vol. 18, No. 8, pp. 727-733, 1999.