Cryptanalysis of Khan's remote user authentication scheme

Chin-Ling Chen Wei-Chech Lin Zong-Min Guo
Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taichung, Taiwan 41349, ROC.
clc@mail.cyut.edu.tw; weichech@gmail.com; ckljdstar@gmail.com

Abstract

Khan improved Yang et al.'s authentication scheme of remote user and server authentication problem. He proposed mutual authentication scheme to resist forgery server attack. However, his scheme suffers from reflection attack. Our proposed scheme can avoid leaking of user's information, resist mostly attacks and ensure the security via improved mutual authentication mechanism.

Keywords: mutual authentication, RSA, smart card

1. Introduction

Since the remote user tries to access a service via 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 the 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 whole system [2].

To solve the fault of stolen-verifier attack of the Lamport's scheme, Yang and Shieh [10] 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 [8] pointed out that Chan and Cheng's attack made no sense and has been shown that Yang and Shieh's scheme still suffers from impersonation attack. Afterward, Yang et al. [9] 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.

Unfortunately, Khan [3] showed that Yang et al.'s scheme was still vulnerable to impersonate attack and therefore proposed an improved scheme. That is, their schemes perform unilateral authentication (only user's authentication), and user has no information about whether the authentication server is authentic or not. In Khan's scheme, he performs the mutual authentication technique to mend the flaws (such as server spoofing attack and server verifiable) aim at the security of Yang et al.'s scheme. However, in this paper, we demonstrate that Khan's scheme is still vulnerable and can be easily attacked. Consequently, we also propose an improved scheme to enhance the security.

The rest of the paper is organized as follows: In section 2, we briefly review of Khan's mutual authentication scheme. In section 3, we elaborate the cryptanalysis of their scheme. In section 4, we present an enhance security scheme on Khan's scheme. In section 5, we analyze and make a comparison with related works. Finally, we conclude this paper in section 6.

2. Review of Khan's scheme

Khan proposed timestamp-based and nonce-based password authentication scheme for synchronous and non-synchronous network, respectively. In the following subsections, we review both of their schemes.

The Khan's notation:

ID_i:	The i^{th} remote user's identity.
PW_i:	The i^{th} remote user's password.
U_i:	The i^{th} user
CID_i	A smart card's identifier.
$h(\cdot)$:	A collision-resistant one-way hash function.
$\oplus$:	Exclusive-or operation.
N:	A nonce.
r_i:	The i^{th} random number.
$\|$:	The concatenation operation.

2.1 Timestamp-based password authentication scheme

In the timestamp-based scheme, there is an existence of key information center (KIC) which manages the registration of the users, and issues a smart card. This scheme is composed of the three phases: registration, login and authentication phase.

2.1.1 Timestamp-based registration phase

In the registration phase, user U_i chooses his or her ID_i and password PW_i, then sends these information to the KIC via secure channel. The system of the KIC performs the following operations:

Step 1: Generates two large prime numbers p and q, and computes $n = p \cdot q$.

Step 2: Chooses a prime numbers e and an integer d which satisfies $e \cdot d \equiv 1 \bmod (p-1)(q-1)$, where e and d are public and private keys of the KIC respectively.

Step 3: Finds an integer g, which is a primitive element in both GF(p) and GF(q), where g is the system's public information.

Step 4: Generates a smart card's identifier CID_i for the user and computes $S_i = ID^{CID_i \cdot d} \bmod n$ and $h_i = g^{PW_i \cdot d} \bmod n$.

Step 5: At the end of the registration phase, KIC stores $n, e, g, ID_i, CID_i, S_i$ and h_i into the smart card and issues card to the U_i through secure channel.

2.1.2 Timestamp-based login phase

In the login phase, user inserts his smart card into the input device and enters his ID_i and PW_i, and then the smart card performs the following procedures:

Step 1: Generates a random number r_i and computes $X_i = g^{PW_i \cdot r_i} \bmod n, Y_i = S_i \cdot h_i^{r_i \cdot T} \bmod n$, where T is the current timestamp of the input device.

Step 2: U_i sends the login message $M = (ID_i, CID_i, X_i, Y_i, n, e, g, T)$ to the remote server S_j for authentication.

2.1.3 Timestamp-based mutual authentication phase

After receiving the login message from the user, the remote server performs the following procedures:

Step 1: Check whether the format of ID_i and CID_i are correct. If the format is not correct, remote server rejects the login request.

Step 2: If $(T' - T'') \geq \Delta T$, where ΔT denotes the expected valid time interval for transmission delay and T' is the received login message timestamp, the server rejects the login request; otherwise the server acquires its current timestamp T'' and computes $C_1 = h(ID^{CID_i \cdot d} \bmod n \oplus T'')$. Then the remote server sends mutual authentication message (C_1, T'') to the U_i.

Step 3: After receiving the mutual authentication message (C_1, T''), U_i verifies if $(T''' - T'') \geq \Delta T$, where T''' is the current timestamp of the U_i; if it is true, U_i rejects this message.

Step 4: U_i computes $C_1^* = h(S_i \oplus T'')$ and $S_i = ID_i^{CID_i \cdot d} \bmod n$. Afterward, the U_i will check the equation $C_1^* \stackrel{?}{=} C_1$, if it holds, U_i believes that the responding party is authentic system and mutual authentication between U_i and remote system is completed, otherwise U_i terminates this request.

2.2 Nonce-based password authentication scheme

The nonce-based authentication scheme is similar to mutual authentication of timestamp-based password authentication scheme except a nonce N is used to instead of the timestamp.

2.2.1 Nonce-based registration phase

This scheme is the same as the registration phase of timestamp-based password authentication scheme.

2.2.2 Nonce-based login and authentication phase

Step 1: The smart card sends a message

$M_1 = (ID_i, CID_i)$ to the server. When receiving the message M_1 from the user, the server checks whether the ID_i and CID_i are correct or not; if it is true, the remote server generates $N = h(r_j)$ and sends N to the user.

Step 2: Once U_i receiving the nonce N, he or she computes the following equations:

$$X_i = g^{PW_i \cdot r_i} \bmod n$$
$$Y_i = S_i \cdot h_i^{r_i \cdot N} \bmod n$$
$$M_2 = (X_i, Y_i, n, e, g)$$

where S_i is stored in the user's smart card and r_i is a random number. Then user sends M_2 to the server.

Step 3: After receiving the message M_2, the server checks $(Y_i)^e \stackrel{?}{=} ID_i^{CID_i} \cdot X_i^N$ and computes $C_1 = h(ID_i^{CID_i \cdot d} \bmod n \oplus N)$. Then server sends message C_1 to the U_i.

Step 4: Then U_i computes $C_1^* = h(S_i \oplus N)$ and compares whether $C_1^* \stackrel{?}{=} C_1$ or not. If it holds, U_i believes that the responding party is authenticated server, and mutual authentication between U_i and remote server is completed; otherwise the login request will be rejected.

3. Cryptanalysis of Khan's scheme

Both of Khan's timestamp-based and nonce-based password authentication schemes are vulnerable to some attacks. In the following subsection, we show the security faults of his scheme.

3.1 Vulnerable to attacks

3.1.1 Denial of service (DoS) attack

In Khan's nonce-based password authentication scheme, the adversary is able to impersonate as the server and to defraud the U_i. Suppose an adversary has intercepted the login message $M_1 = (ID_i, CID_i)$ in a previous login phase, he or she can therefore make the bogus hash value N^* up and send it back to the U_i. Afterward, the U_i uses the responding hash value N^* to generate several variable such as X_i, Y_i and the digest message $M_2 = (X_i, Y_i, n, e, g)$ and then sends M_2 to the server without checking the correctness of the server's identity. Since the adversary can easily to compute the verifier $C_1 = h(ID_i^{CID_i \cdot d} \bmod n \oplus N)$ and then sends the C_1 back to the U_i. The U_i will generate C_1^* and check the legality of the server's identity as follows.

$$C_1^* = h(S_i \oplus N) \stackrel{?}{=} C_1$$

Even if the above verification equation does not hold as the adversary has no knowledge of the private key d. But the adversary has achieved his or her purposes of disturbance and impersonation.

3.1.2 Reflection attack

In Khan's nonce-based password authentication scheme, if an adversary has intercepted and blocked the message transmitting in login phase, i.e. $M_1 = (ID_i, CID_i)$, he can impersonate server to send $N = f(r_i)$ to U_i. Upon receiving the message N, U_i computes X_i, Y_i and M_2. Without verifying the legality of the server, U_i will be fooled into believe that the adversary is the server. Since U_i cannot actually authenticate the server, Khan's scheme fails to prove the correctness of each identity as they claimed. Therefore, the reflection attack may result from serious problem in Khan's proposed scheme.

3.2 Non-anonymity

Consider the scenarios of an identity disclosure attack where an adversary intercepts the communication messages between U_i and the server, and then try to find the identity of user U_i. In both Khan's timestamp-based and nonce-based remote user authentication schemes, the login message M_1 includes of U_i's identity ID_i and sends in plaintext. Therefore, all users' identities are known to all users, and Khan's scheme cannot meet the requirement of anonymity as require in remote user authentication scheme.

3.3 Unable to mutual authentication

In case of Khan's timestamp-based remote user authentication scheme, the both proof authentication equations $(Y_i)^e$ and C_1^* will be verified in both the server side and the user side as follows.

$$(Y_i)^e \stackrel{?}{=} ID_i^{CID_i} \cdot X_i^T \bmod n$$
$$C_1^* = h(S_i \oplus T'') \stackrel{?}{=} C_1$$

If the responding message C_1 is unfortunately

intercepted by an adversary, then the server may not know the situation of losing the message.

And there also not suitable in case of Khan's nonce-based remote user authentication scheme, only a random number N is generated by the server in the authentication phase. And the random number N is used to provide the certification of the server's identity, and the computations are as follows.

$$N = h(r_i)$$
$$Y_i = S_i \cdot h_i^{r_i \cdot N} \bmod n$$
$$(Y_i)^e \stackrel{?}{=} ID_i^{CID_i} \cdot X_i^N$$
$$C_1 = h(ID_i^{CID_i \cdot d} \bmod n \oplus N)$$
$$= h(S_i \cdot N)$$
$$= C_1^*$$

done, there may result in a potential problem in repudiation. Therefore, the property of mutual authentication which Khan has claimed will face on the challenge.

4. Our improved scheme

To prevent the potential risk described above in Khan's scheme, we propose an improvement scheme aims to enhance the security between remote user and the server. Due to the timestamp-based scheme will face the issue of time-synchronization. Therefore, we don't discuss the timestamp-based scheme here, but only improves the nonce-based scheme of Khan's proposed schemes. The scenarios of our proposed improvement scheme are illustrated in Fig. 1.

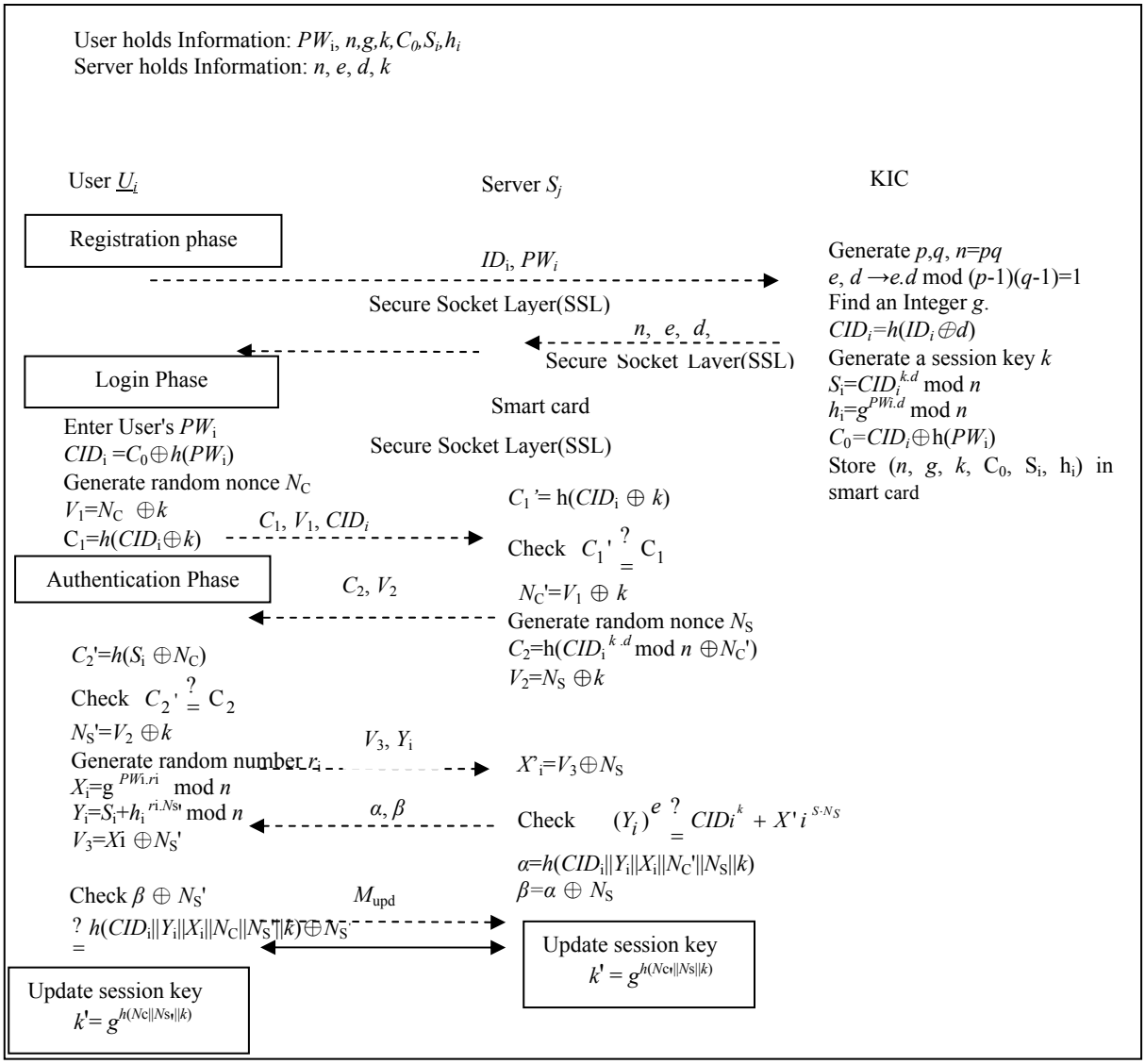

Fig. 1. Our improved nonce-based password authentication scheme

Since the random number N doesn't update when the authentication session has not been

Our scheme divides into four phases namely

system initialization, user registration phase, login phase and authentication phase. We describe the notations and the steps of each phase as follows.

4.1 Notations

All the entities involved in our protocol are called parties and communicate through remote computer networks. The notations U_i, S_j, (n, e), d are used to denote the names of the parties client, remote server, the RSA public key [6], and the corresponding RSA private key, respectively. 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
CID_i:	The dynamic authenticator of the i^{th} user.
k:	A session key.
$h(\cdot)$:	A collision-resistant one-way hash function.
$\oplus$:	Exclusive-or operation.
N_x:	A random nonce x.
r_i:	The i^{th} random number.
$\|$:	The concatenation operation.
$A \stackrel{?}{=} B$:	Compare whether A equals to B or not.

4.2 System initialization and user registration phase

In our scheme, a 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:

(1) Two randomly and independently large prime numbers p and q.
(2) A RSA modulus $n = p \cdot q$.
(3) A generator g which is the primitive element of GF(p) and GF(q).
(4) A collision-resistant hash function $h(\cdot)$ (where $h(\cdot)$ is either SHA-1 or MD5 hash functions[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 session k with the server S_j for a login proof.

When the user registers as the client (using a nickname instead the real identity to protect one's privacy), one KIC must be generated and published the necessary parameters for every nickname assigned to the user as follows.

$$sig_i = CID_i^{k \cdot d} \bmod n$$
$$CID_i = h(ID_i \oplus d)$$
$$C_0 = CID_i \oplus h(PW_i)$$

Then the KIC stores the verifiable information $(n, g, C_0, k, sig_i, h_i)$ into the smart card.

4.3 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: $U_i \rightarrow S_j : CID_i, C_1, V_1$

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

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

Afterward, the U_i will generate a nonce N_c and performs the following operations with the session key k to provide the certificate to the S_j.

$$V_1 = N_C \oplus k$$
$$C_1 = h(CID_i \oplus k)$$

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

4.4 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: $S_j \rightarrow U_i : C_2, V_2$

To verify the correctness of the received login message, the server S_j computes C_1' with the session key k and the received (C_1, V_1, CID_i) and then compares with the received C_1 as follows.

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

Checks $C_1' \stackrel{?}{=} C_1$

If the above equation holds, the S_j proceeds to acquire the nonce N_C'.

$$N_C' = V_1 \oplus k$$

Afterward, the S_j generates a nonce N_S and computes the response message (C_2, V_2) back to the U_i with the private key d and session key k as follows.

$$C_2 = h(CID_i^{k \cdot d} \bmod n \oplus N_C')$$
$$V_2 = N_S \oplus k$$

Otherwise, reject 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' and verifies its correctness by checking whether C_2' equals to the received C_2 or not. The computing equations are as follows.

$$C_2' = h(S_j \oplus N_C)$$

Checks $C_2' \stackrel{?}{=} C_2$

If the above equation holds, the U_i proceeds to acquire the nonce N_S' with his or her session k and the received V_2.

$$N_S' = V_2 \oplus k$$

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.

$$X_i = g^{PW_i \cdot r_i} \bmod n$$
$$Y_i = S_i + h_i^{r_i \cdot N_S'}$$
$$V_3 = X_i \oplus N_S'$$

Finally, the authentication procedure has been done and the U_i will send the message (V_3, Y_i) to the server S_j to ask 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.

$$X_i' = V_3 \oplus N_S$$
$$(Y_i)^e \stackrel{?}{=} CID_i^{k \cdot d} + X_i'^{N_S}$$

If the above equation holds, the S_j will continuously generate the confirmation message (α, β) and send it back to the U_i. The computation equations are shown as below.

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

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

After receiving the confirmation message (α, β), the U_i will check its correctness with the serial variable as he knows in advance.

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

If the above equation holds and to avoid the potential attacks, the U_i will continuously regenerate the session key k' and send the updated request message M_{upd} back to the S_j. The computation of k' is shown as below.

$$k' = g^{h(N_C \| N_S' \| \alpha)}$$

Step 5: $U_i \longleftrightarrow S_j$:

Upon receiving and verifying the validity of the updated request message M_{upd}, the S_j computes the newly session key k' and executes the procedure of replacing the session key k with k'.

$$k' = g^{h(N_C' \| N_S \| \alpha)}$$

Thus, both of requirements of mutual authentication and session key agreement can therefore be achieved..

5. Security analyses of our improved scheme

5.1 Against various attacks

5.1.1 DoS attack issue

Since the adversary may resends the previous login messages (C_1, V_1, CID_i) and expects to pass the verify procedure of S_j. 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 session key k as shown in below:

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

Checks $C_1' \stackrel{?}{=} C_1$

And the session key k will be updated when an authentication phase has been done. The equation is as follows:

$$k = g^{h(N_C \| N_S' \| \alpha)}$$

Thus, it is infeasible to the adversary to palsy our scheme by sending the illegal login message unceasingly. Therefore, our proposed scheme can against the DoS attack.

5.1.2 Replay attack issue

In authentication phase, the adversary may play a replay attack by resending the authenticate message and could be succeeded whenever the nonce used between the communication parties is unchangeable. Nevertheless, all the nonce (i.e.,

N_C and N_S) are variable and would be verified by another party during the communication in our proposed scheme. The examination equation is as follows.

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

$$(Y_i)^e \stackrel{?}{=} CID_i^{k \cdot d} + X_i'^{N_S}$$

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

5.1.3 Forgery attack issue

Most of the transaction message of our proposed scheme contains $(C_1, V_1, C_2, V_2, V_3, Y_i \text{ and } \alpha)$. If an adversary expects to forge a legal message, it is necessary to get the session key k. Since the session key k has only shared between the communication parties and is protected under the collision-resistant hash function $h(\cdot)$. Therefore, it is computing infeasible to the adversary to extract the session key k directly.

5.2 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 CID_i or C_0 as follows.

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

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

Therefore, the property and security of anonymity in our improved scheme can easily be achieved.

5.3 Mutual authentication issue

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.

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

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', Y_i as follows.

$$X_i' = V_3 \oplus N_S$$

$$(Y_i)^e \stackrel{?}{=} CID_i^{k \cdot d} + X_i'^{N_S}$$

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.

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

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

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

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

Finally, both of the U_i and S_j computes the newly session key k' and replaces the old session key k as follows.

$$k' = g^{h(N_C \| N_S' \| k)} = g^{h(N_C' \| N_S \| k)}$$

Therefore, from the above analyses, it is clearly that our scheme can reaches the purpose of mutual authentication by maintaining the verifiable of the proofs.

5.4 Two-factor issue

If both 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 stolen, but not both. This security property is called two-factor security. For our improved scheme, the parameters (n, g, C_0, k, 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. Though 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 be extracted 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 or she still cannot obtain PW_i directly. Thus, our scheme indeed provides two-factor security.

Table 1. the comparisons of our proposed scheme and previously proposed schemes.

	Yang et al.'s [9]	Kim et al.'s[4]	Khan[3]	Our scheme
Reflection attack	N	N	N	Y
DoS attack	N	N	N	Y
Early detection	N	N	N	Y
Mutual authentication	N	N	N	Y
Anonymity	N	N	N	Y
Parallel session key	N	Y	Y	Y
Leak of password	Y	Y	Y	N

5.5 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 they claim. As well as the proposed scheme construct the session key implicitly on performing user identification, requiring no extra overhead. In addition, the proposed scheme further provides parallel session key confirmation between each party.

6. Conclusion

We have shown that Khan's remote user authentication scheme is vulnerable to reflection attack and Dos attack, and is not anonymity. Additionally, we have proposed an improved scheme, in which the serial of attacks are prevented and the anonymity is achieved. Besides, we make a comparison with previous schemes. It is clearly that our scheme can resist mostly attacks and support the security. In the future, we hope that this remote authentication technique can be widely adopted and expanded in smart card-based or mobile device-based schemes.

References

[1] Chan, C. K. and Cheng, L. M., "Cryptanalysis of a Timestamp-Based Password Authentication Scheme," ***Computers & Security***, Vol. 21, No.1, pp. 74-76, 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] Rivest, R. L., Shamir, A. and Adleman, L., "A method for obtaining digital signatures and public-key cryptosystems," ***ACM Communications***, Vol. 21, No.2, pp. 120-126, 1978.

[7] Sarkar, P., "Domain extender for collision resistant hash functions: Improving upon Merkle–Damgard iteration," ***Discrete Applied Mathematics***, 2008.

[8] 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.

[9] 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.

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