International Journal of Security and Its Applications
Vol. 10, No. 12 (2016), pp.311-322
http://dx.doi.org/10.14257/ijsia.2016.10.12.25

Cryptanalysis of a Multiple Server Smart Card based Authentication Scheme

Kwang Cheul Shin

Division of Industrial Management Engineering, Sungkyul University, Sungkyul University-Ro 53, Anyang 8 dong, Manan-gu, Anyang-si, Gyeonggi-do 14097, Korea
skcskc12@sungkyul.ac.kr

Abstract

The design of the remote user authentication scheme for access to the service server is a very important issue in a multi-server environment. In particular, it is essential for mutual authentication and key agreement process between the user and the server. Proposed so far many schemes are focused to increase safety, reduce the communication time and calculation cost. On the other hand, there is a limit to overcome a variety of attacks. In recent years, Jain et al'.s proposed the authentication scheme, such as mutual authentication and session key establishment, smart card-based multi-server authentication scheme to withstand a variety of attacks. In this paper, I analyze that Jain et al'.s scheme is not secure against user impersonation attack, man-in-the-middle attack, DoS attack, reply attack etc.

Keywords*: Smart card, Impersonation attack. Mutual authentication, man-in-the-middle attack*

1. Introduction

Today, many network services are needed to authenticate the user without regard to location and time. Mainly, user authentication is used in online games, on-line shopping, on-line reservation by the application server. Authentication requires a pre-registered in order to prevent the non-authorized access to the server. Users are required to register previously. Traditional single-server authentication method is used for convenience of password authentication system by a number of web services.

The server has a user identifier and registers the user's password to the verification table for confirmation [1-4].

If the traditional password authentication method is applied to a multi-server environment, the user performs the registration procedure many times and results in a high overhead at the registration center and the network. Each network user has to remember the ID of different identifiers and corresponding password, when the user must remember each time the login to a variety of remote server. This is a problem that the management of secret information is shared between the participants.

Therefore, in the multi-server environment that it is difficult to apply the conventional authentication scheme, because it must be registered several times in different remote servers stores a different ID and password. In addition, the traditional password-based remote user authentication schemes are still easily broken by simple dictionary attacks due to the low entropy of password and the secret information stored in smart card that it could be extracted by physically monitoring power consumption [5].

To eliminate such problem, a password authentication scheme has been proposed base upon smart card. Smart card is tamper resistant integrated circuit card with memory and processor capable of performing computations. In this direction, many attractive authentication schemes have been proposed using smart cards during the last decade.

ISSN: 1738-9976 IJSIA

A remote user authentication scheme based on symmetric key cryptosystem of Juang[6] was proposed in 2004. But his scheme was found susceptible to the insider attack.

In 2008, Tsai [7] proposed authentication scheme without using verification table based on hash function and smart card. A common feature in most of conventional multi-server authentication scheme is as follows. The user used of the fixed identifier. Therefore, adversary may collect partial authentication information. Adversary is identifies the same user and obtains transmission information to specify the target of a variety of attacks.

2009, Liao *et al*'s [8] proposed a dynamic remote user ID authentication scheme for anonymous users, as well as using only one-way hash function in a multi-server environment in order to avoid a hazard. But Hsiang *et al*'s [9] pointed out that Liao *et al*'s scheme is vulnerable to insider attacks, spoofing attack, server forgery attack, registration center forgery attacks. and they proposed an improved scheme.

In 2011, Sood *et al*'s [10] has pointed out that Hsiang *et al*'s scheme is not secure.

They found that the Hsiang *et al*'s scheme is vulnerable to replay attacks, spoofing attacks, smart card stolen attacks, password changes attack. and they are proposed new scheme to prevent their anonymity and several other attacks. However, Sood *et al*'s scheme is possible impersonation attack as a legitimate user logged into the system when the smart card is lost.

In 2012, Tsaur *et al*'s scheme [11] proposed a mutual authentication and key agreement scheme in a multi-server environment and in 2013, Xu *et al*'s[12] proposed a scheme that can dynamically change the user identifier of each session. However, when user using the services server, there is a problem that user must be re-registered again.

In 2014, Jain *et al*'.s[13] proposed an efficient and robust multi-server authentication scheme using smart card. Security of this scheme depends upon cryptographic one-way hash function. Users should be able to access all of the resources in the server once registered in a multi-server environment. Mutual authentication and key agreement process is essential between user and server. And it must be able to withstand a variety of attacks that occurred by the adversary.

In this paper, I analysis that Jain *et al*'.s scheme is not secure against a server impersonation attack, user impersonation attack, man-in-the-middle eavesdrop attack, smart card stolen attack. The rest of the paper is organized as follows : Section 2 reviews the Jain *et al*'.s scheme. In section 3, I show how to attack Jain *et al*'.s scheme. Finally, I make a conclusion in section 4.

2. Review of Jain *et al*.'s Authentication Scheme

This section reviews a user authentication scheme proposed by Jain *et al*[13]. There are three communication parties in Jain *et al*'.s scheme: User i, Registration Center(under RC), Service Application Server S. This scheme is composed of four phases : registration phase, login phase, authentication-key agreement phase, and password change phase. I describe each phase in detail in 2.1-2.4, and Table 1 shows the notations used in the remainder of the paper.

2.1 Registration Phase

This phase is divided into two sub-phase: Server Registration phase and User Registration phase(as shown in Fig. 1).

Table 1. Notation and Description

Notation	Description
RC	Registration Center
U_i	i^{th} remote user
ID_i	Identity of user i
pw_i	Password of user i
$h(\cdot)$	An one-way hash function
S_j	j^{th} authentication server($1 \le j \le n$)
SID_j	Identity of user S_j
x	Secret key of Registration Center
d	Secret number of Registration Center
$\|$	Concatenation operation
$\oplus$	XOR operation
$\rightarrow$	Insecure channel
$\cdots>$	Secure channel

2.1.1 Server Registration Phase

In this phase, while application server S_j want to become an authorized server. S_j select SID_j and submits it to RC over a secure channel. Upon receiving the registration request from S_j, RC computes the server secret parameter $SS_j=h(SID_j \| h(x)) \oplus h(d)$ and sends $<SS_j, h(d)>$ to S_j through a secure channel.

The authorized server uses the SS_j and h(d) to check the user's legitimacy in authentication phase.

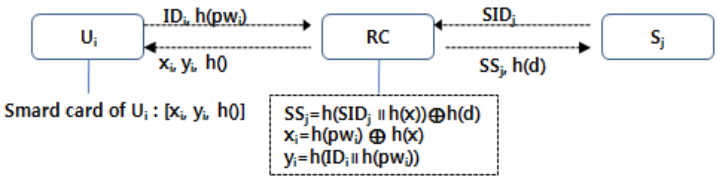

Figure 1. User and Server Registration Phase

2.1.2 User Registration Phase

The registration phase of Jain *et al*'.s scheme is described in Fig 1. User i needs to perform the user registration phase with the registration center using a secure channel.

U_i selects ID_i and pw_i, compute $h(pw_i)$ and submits $<ID_i, h(pw_i)>$ to RC over a secure channel. Once the registration request is received, RC computes $xi=h(pw_i) \oplus h(x)$, $yi=h(ID_i \| h(pw_i))$ and issues a smart card over secure channel to U_i by storing $<x_i, y_i, h()>$ into smart card memory.

2.2 Login Phase

The Login and Authentication phase for the proposed scheme are described in Fig 2. In this phase, U_i inserts the smart card to the card reader and keys in ID_i and pw_i'.

The smart card computes $y_i'=h(ID_i \| h(pw_i'))$ and verifies computed y_i' equals stored y_i or not. If not, the user terminates the session. If true, reader generates a random number nonce N_1, the smart card computes the following :

$a_i=x_i \oplus h(pw_i)$

$b_i=h(SID_j \| a_i)$

$c_i=h(b_i \| N_1)$

And U_i sends the login request <ID_i, SID_j, N_1, c_i> to S_j.

2.3 Authentication Phase

Upon receiving the login requests <ID_i, SID_j, N_1, c_i>. S_j first checks the validity of ID_i to accept/reject the login request. If true, S_j computes $c_i'=h((SS_j \oplus h(d)) \| N_1)$ and then checks whether computed c_i' equals received c_i or not. If it holds, S_j generates a nonce N_2, the S_j computes the following :

$SKey_{ji}=h(ID_i \| SID_j \| (SS_j \oplus h(d)) \| N_1 \| N_2)$

$d_i=h(SKey_{ji} \| N_2)$

And S_j sends the message <ID_i, d_i, N_2> to U_i.

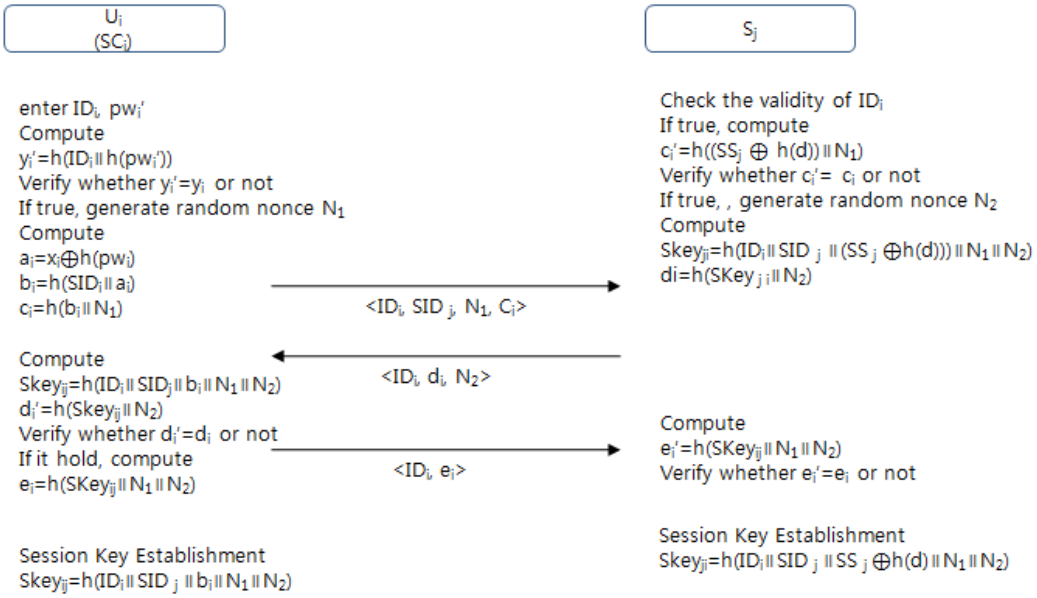

Figure 2. Login and Authentication Phase

As it described in Figure 2, after getting the message <ID_i, d_i, N_2> from S_j, U_i computes the following :

$SKey_{ij}=h(ID_i \| SID_j \| b_i \| N_1 \| N_2)$

$d_i'=h(SKey_{ij} \| N_2)$

And checks whether the computed d_i' equals received d_i or not. If it holds, S_j is authentic otherwise terminate the session.

Subsequently, U_i computes $e_i=h(SKey_{ij} \| N_1 \| N_2)$ and send <ID_i, e_i> to S_j. Once the message <ID_i, e_i> is received, S_j computes $e_i'=h(SKey_{ji} \| N_1 \| N_2)$ and checks whether computed ei equals received e_i or not. If it holds, mutual authentication is achieved. Both the parties agree upon a common shared session key. The session key $SKey_{ij}$ is generated by the user i and the session key $SKey_{ji}$ generated by the Server j.

$SKey_{ij}=h(ID_i \| SID_j \| bi \| N_1 \| N_2)$

$SKey_{ji}=h(ID_i \| SID_j \| (SS_j \oplus h(d)) \| N_1 \| N_2)$.

2.4 Password Change Phase

The proposed password change phase is executed when the user i wants to update his password. In this phase, The user i can easily change his password without any assistance from the registration center. Ui insert the smart card to the card reader and keys in ID_i and pw_i'. After this, the smart card computes $y_i'=h(ID_i \| h(pw_i'))$ and verifies whether

computed y_i' equals stored y_i or not. If true, U_i enters a new password pw_{inew}. The smart card computes $x_{inew}=x_i \oplus h(pw_i) \oplus h(pw_{inew})$, $y_{inew}=h(ID_i \| h(pw_{inew}))$ and stores x_{inew}, y_{inew} instead of x_i, y_i respectively in the smart card memory.

3. Security Analysis of Jain *et al*.' Scheme

In this section, I analyze the security of Jain *et al*'.s scheme. In Jain *et al*'.s scheme, they claimed as follows: their scheme provides security against server impersonation attack, user impersonation attack, reflection and parallel session attacks, replay attack, insider attack, password guessing attack, stolen verifier attack and smart card loss attack.

Unfortunately, I found that their scheme still has a many vulnerabilities. Any registered but malicious user can derive the session key between any user and server by eavesdropping their communication information in public channel. Also, this scheme cannot checks the freshness of login message, so it is use that the malicious user sends the previous message to the server. Therefore, malicious user can success the reply attack and DoS attacks.

3.1 User impersonation attack

User i send the login message $<ID_i, SID_j, N_1, c_i>$ to the server j in order to use the server service. Then user A is a legitimate registered user malicious. Malicious user A intercepts the login message $<ID_i, SID_j, N_1, c_i>$, and he/she can completely spoofing the server j.

Malicious users A have $x_a=h(pw_a) \oplus h(x)$, $y_a=h(ID_a \| h(pw_a))$ that it is provided by registration center, and it is stored on the smart card.

Also, malicious user A has the parameters h(x) derived from $x_a=h(pw_a) \oplus h(x)$.

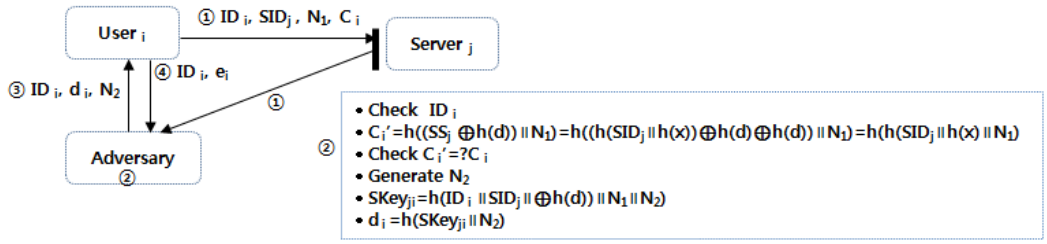

Figure 3. User impersonation attack scenario

User impersonation attack is the same as Figure 3, the scenario is as follows.

---

step 1 The legitimate user i sends a login message $<ID_i, SID_j, N_1, c_i>$ to server j.

step 2 Legally registered malicious user A will intercept the login message $<ID_i, SID_j, N_1, c_i>$ and the following operations.

$c_i'=h(h(SID_j \| h(x) \| N_1)$

$h(h(SID_j \| h(x) \| N_1))$ is the same as $h((SS_j \oplus h(d)) \| N_1)$, and also $h((SS_j \oplus h(d)) \| N_1)$ is the same as $h((h(SID_j \| h(x)) \oplus (d) \oplus (d)) \| N_1)$.

Next generate a nonce N_2

$SKey_{ji}=h(ID_i \| SID_j \| (h(SID_j \| h(x) \| N_1 \| N_2)$

$h(ID_i \| SID_j \| (h(SID_j \| h(x) \| N_1 \| N_2)$ is the same as $h(ID_i \| SID_j \| (SS_j \oplus h(d)) \| N_1 \| N_2)$.

$d_i=h(SKey_{ji} \| N_2)$

step 3 The malicious user A sends $<ID_i, d_i, N_2>$ to U_i.

step 4 After getting the message $<ID_i, d_i, N_2>$ from malicious user A, User i computes the following :

$SKey_{ij}=h(ID_i \| SID_j \| b_i \| N_1 \| N_2)$

$d_i'=h(SKey_{ij} \| N_2)$

Checks whether computed d_i' equals received d_i or not. If it holds, User i computes e_i the following :

$ei=h(SKey_{ij} \| N_1 \| N_2)$

step 4 User i send $<ID_i, e_i>$ to malicious user A.

In this proposed Jain *et al*'.s scheme, the login request contains $<ID_i, SID_j, N_1, c_i>$. It contains $c_i=h(b_i \| N_1)=h(h(SID_j \| a_i) \| N1)=h(h(SID_j \| x_i \oplus h(pw_i)) \| N_1)$. In order to securely perform impersonation attack, the attacker needs to guess the correct values of $h(pw_i)$ and $h(x)$. But, malicious user A is not necessary to use $h(pw_i)$ owned by the user. The reason is as follows.

$c_i=h(b_i \| N_1)$

$=h(h(SID_j \| a_i) \| N_11)$

$=h(h(SID_j \| x_i \oplus h(pw_i)) \| N_1)$

$=h(h(SID_j \| h(pw_i) \oplus h(x) \oplus h(pw_i)) \| N_1)$ //$x_i=h(pw_i) \oplus h(x)$

$=h(h(SID_j \| h(x) \| N_1)$

The malicious user A does not know the $h(pw_i)$ of the user i. But he can calculate the c_i.

In addition, he can calculate the session key $SKey_{ij}$, and he can communicate as if the server j. Therefore, Jain *et al*'.s scheme is vulnerable to user impersonation attack.

3.2 Man-in-the-middle Attack

Jain *et al*'.s scheme allows a legal malicious user to eavesdrop the communication between a user i and a server j and then acquire communication information that they are transmitted over the communication once the eavesdropping succeeds.

Two entities(user i and server j) presume that they are successfully connected to peer side. However, they are actually connected to an intermediate entity so that the intermediate entity can collect the data sent by the user i and transmit the data to server j after manipulating it.

Man-in-the-middle attack is the same as Figure 4, the scenario is as follows.

---

step 1 The legitimate user i sends a login message $<ID_i, SID_j, N_1, c_i>$ to server j.

step 2 Legally registered malicious user A will intercept the login message $<ID_i, SID_j, N_1, c_i>$ and response message $<ID_i, d_i, N_2>$, He can calculate the session key as follows.

From the $SKey_{ji}=h(ID_i \| SID_j \| (SS_j \oplus h(d)) \| N_1 \| N_2)$

$(SS_j \oplus h(d))$ is $h(SID_j \| h(x))$

Using the equation $h(SID_j \| h(x))$ and calculates a session key as follow.

$SKey_{ji}=h(ID_i \| SID_j \| h(SID_j \| h(x) \| N_1 \| N_2)$

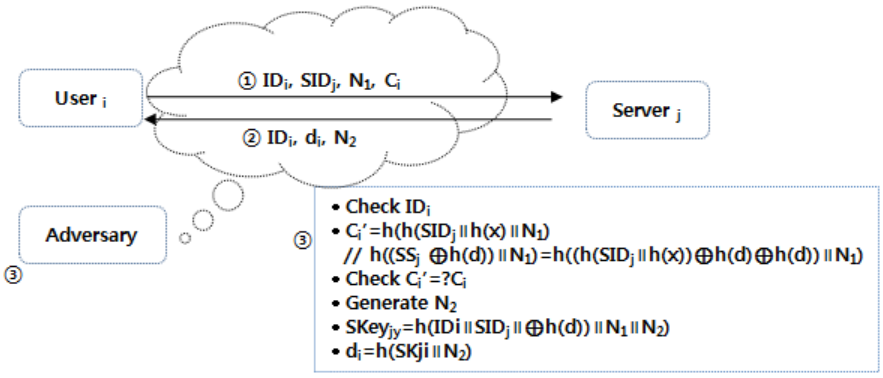

Figure 4. Man-in-the-middle Attack Scenario

Since malicious user A has intercepted the messages transmitted and received between the user and the server j. And he/she can decrypt by the session key from all encrypted messages. Therefore, Jain *et al*'s scheme is vulnerable to man-in-the-middle attacks.

3.3 Denial of Service Attack

Figure 5 describes DoS attack on Jain *et al*'.s scheme.

The malicious user can attempt to make the server or network resource unavailable if he use a many intercepted authentication message.

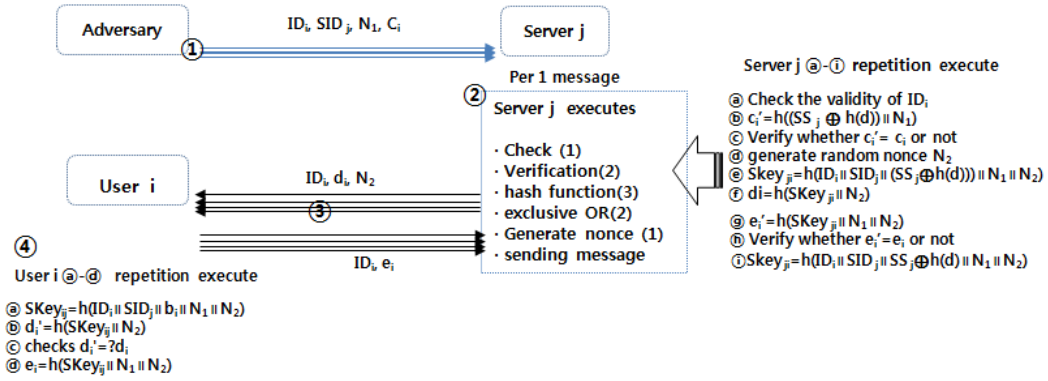

Figure 5. Denial of Service Attack Scenario

Denial of Service attack is the same as Figure 5, the scenario is as follows.

---

step 1 The malicious user gets the previous message ① <ID_i, SID_j, N_1, c_i> from a legitimate user and sends that it is mass copy without modifying the login message to the server j.

step 2 Then, the server j executes operation ② and sends message ③ to the user i.

The processes of operation ② include check the validity of ID_i and verification 2 times, executing the hash function 3 times, calculating the exclusive-or operation 2 times, and generation a random nonce.

step 3 The user i repeatedly executes the message ④ .

step 4 The user i send a message <ID_i, e_i> of n times to S_j.

step 5 <ID_i, e_i> is received, S_j computes n times the following :

$e_i'=h(SKey_{ji} \| N_1 \| N_2)$

checks $e_i'=?e_i$

$SKey_{ij}=h(ID_i \| SID_j \| b_i \| N_1 \| N_2)=h(ID_i \| SID_j \| (SS_j \oplus h(d)) \| N_1 \| N_2)$.

In Jain *et al*'s scheme, the server j does not check whether login message sent by the user i is freshly created or not.

Even if the third party sends message which has been intercepted in past to the server j, the server j cannot say that it is a message which has been created in past or in present. Therefore, Jain *et al*'.s scheme is vulnerable to Dos and replay attacks.

3.4 Reply Attack

The Jain *et al*'.s scheme is proposed that can prevent the replay attack. The reason for this is as follows.

Here, the replay attack will fail because the freshness of the messages transmitted in the login and authentication phases is provided by the random nonce N_1 and N_2. These are generated independently, and their values differ among sessions. So attackers cannot enter the system by re-sending the earlier transmitted messages to pretend to be legal users. However, same as a service denial attack of 3.3, Jain *et al*'.s scheme does not ask the question against the login message whether it is freshly created so as to determine if the message is created in real time. Furthermore, the scheme does not check if user i's parameter nonce N_1 is newly created or not. Therefore, the scheme is vulnerable to a replay attack.

3.5 No Perfect Forward Secrecy

Perfect Forward Secrecy has characteristic that security of session key that are provided in the process of key distribution must not be compromised even if private key is exposed. According to this characteristic, new key information is mathematically not related to previous key information at all. Therefore, even in the event that someone detects previous session key, he/she cannot predict new session using the previous one.

In Jain *et al*'.s scheme, They argued as follows : The session key $SKey_{ij}=h(ID_i \| SID_j \| b_i \| N_1 \| N_2)=h(ID_i \| SID_j \| (SS_j \oplus h(d)) \| N_1 \| N_2)$ is associated with $h(pw_i)$, h(x), h(d) which are unknown to the adversary. Even though the past session key is compromised, the adversary cannot extract these parameters due to the security of one-way hash function. Moreover, it is infeasible to guess these values simultaneously. Thus, the adversary cannot obtain any further session key. However, consider the following scenario.

---

step 1 malicious user A got ID_i, SID_j, N_1 and N_2 in previous public channel.

step 2 malicious user A knew common parameter h(x) of user's from $xi=h(pw_a) \oplus h(x)$

step 3 $SKey_{ji}=h(ID_i \| SID_j \| h(SID_j \| h(x) \| N_1 \| N_2)$

First, the malicious user got ID_i, SID_j, N_1 and N_2 in previous public channel between user i and server j. Next, the malicious user knew common parameter h(x), so the malicious user can calculate $SKey_{ji}$ from ID_i, SID_j, N_1 and N_2.

In Jain *et al*'.s scheme, they claimed that $h(ID_i \| SID_j \| (SS_j \oplus h(d)) \| N_1 \| N_2)$ is associated with (pw_i), h(x) and h(d) which are unknown to the adversary. But because $(SS_j \oplus h(d))$ and $h(SID_j \| h(x)$ are equal, adversary may also be calculated the session key without knowing h(d), $h(pw_i)$. Therefore, his schemes are vulnerable to perfect forward secrecy.

4. Analysis Result of Jain *et al*'.s Scheme

Each users must register with the registration center to access the service server. Also, service server be registered with the registration center. Also, service servers should be registered in the registration center.

The design errors of Jain *et al*'.s scheme is as follows.

First, the user sends the ID_i and $h(pw_i)$ to the registration center. The registration center calculates x_i, x_j, x_k, ...x_n by using its own secret key x.

$x_i=h(pw_i)\oplus h(x)$

$x_j=h(pw_j)\oplus h(x)$

$x_k=h(pw_k)\oplus h(x)$

:

$x_n=h(pw_n)\oplus h(x)$

Therefore, the legitimate all member will have the parameters h(x) in common, they are using the parameter h(x) and also they can be calculated the session key of the other legitimate user.

Table 2. The Re-analysis Result of Security Properties

security components	Analysis of Jain *et al*'.s scheme	Re-analysis Jain *et al*'.s scheme
Session key agreement	Yes	Yes
Resist user impersonation attack	Yes	No
Resist server spoofing attack	Yes	Yes
Resist man-in-the-middle attack	△	No
Resist DoS attack	△	No
Resist reply attack	Yes	No
Session key attack	Yes	No
User anonymity	△	No
Mutual authentication	Yes	Yes
Free from maintaining verification table	Yes	Yes
Perfect Forward Secrecy	△	No

Second, the user should use his password when he generating the login message. However, the password was not used.

In $a_i=xi\oplus h(pw_i)$, user's password is used for calculation. In the long run, the calculation result turns to $a_i=h(x)$, which means that user's password is not necessary in the computation. Therefore, the third member computes $b_i(=h(SID_j\| h(x)))$ using his/her own h(x).

Third, this is trap of SS_j and h(d) submitted to servers by RC.

Since $SS_j=h(SID_j\| h(x))\oplus h(d)$, server j is not aware of h(x). Hence, server spoofing attack is secure. However, when anyone among members eavesdrops login message c_i, he/she can get solution of $c_i=h(h(SID_j\| h(x))\| N_1)$.

At this point, an important thing is that server uses $h((SS_j\oplus h(d))\| N_1)$ to calculate c_i in Jain *et al*'.s scheme. However, since $h((SS_j\oplus h(d))\| N_1)=h(h(SID_j\| h(x))\| N_1)$, secret parameters SS_j and h(d) which are only kept by server are of no use to prevent an impersonation attack attempted by the third party.

Fourth, the scheme does not provide anonymity since all information that can be used to generate session key are exposed to public channel due to user's login message $<ID_i, SID_j, N_1, c_i>$ and its response message $<ID_i, di, N_2>$. Therefore, it is true that any legal member is able to calculate session key ($SKey_{ji}=h(ID_i\| SID_j\| h(SID_j\| h(x)\| N_1\| N_2)$).

Table 2 shows the security components was analyzed by Jain *et al*'.s, Also I show the security components by re-analyzed the Jain *et al*'.s scheme. Mark △ is the security components could not be analyzed by Jain *et al*'.s.

In this paper, I was proved vulnerable to attack your impersonation attack, denial of service attacks, man-in-the-middle attacks, replay attacks, perfect forward secrecy.

5. Conclusion

The user shall have access to all the service servers to a single register in a multi-server environment. In particular, it is essential for mutual authentication and key agreement process between the user and the server. Authentication schemes also have to overcome a variety of attacks that occur in a public channel.

In this paper, I analyze and proved the mutual authentication and key agreement process, the ability to overcome a variety of attacks Jain *et al*'.s multi-server smart card based authentication scheme. I found the result of design errors by analyzing the scheme.

For this reason, My analysis reveals its inherent security vulnerabilities, such as user spoofing, man-in-the-middle attacks, no perfect forward secrecy.

References

[1] L. Lamport, “Password authentication with insecure communication”, Communications of the ACM, vol. 24, no.11**, (1981),** pp.770-772.
[2] C. Chang, K.F. Hwang, "Some Forgery Attacks on a Remote User Authentication Scheme Using Smart Cards", Informatics Vol. 14, no. 3, **(2003),** pp.289-294.
[3] M.L. Das, Saxena, A. V.P. Gulati, "A dynamic ID-based remote user authentication scheme", IEEE Trans. Consum. Electron. Vol. 50, no. 2, **(2003),** pp.629-631.
[4] C. I. Fan, Y. C. Chan and Z. K. Zhang, “Robust remote authentication scheme with smart cards”, Computers & Security, vol. 24, , **(2005),** pp.619-28.
[5] T. S. Messerges, E. A. Dabbish, and R. . . Sloan, Examining smart-card security under the threat of power analysis attacks, IEEE Transactions on Computers, vol. 51, no. 5, , **(2002),** pp. 541–52.
[6] Juang W. S. Efficient multi-server password authenticated key agreement using smart cards. Consumer Electronics, IEEE Transactions on, 50(1), **(2004),** 251–.255.
[7] Tsai J. L. Efficient multi-server authentication protocol based on one-way hash function without verification table. Computers & Security, 27(3), **(2008),** 115–121.
[8] Y. P. Liao and S. S. Wang, “A secure dynamic ID based remote user authentication scheme for multi-server environment,” *Computer Standards and Interfaces*, vol. 31, no. 1, **(2009),** pp. 24–29.
[9] H. C. Hsiang, and W. K. Shih, Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment, Computer Standards & Interfaces, Vol. 31, **(2009),** pp. 1118-1123.
[10] S. K. Sood, A. K. Sarje, and K. Singh, 2011 A secure dynamic identity based authentication protocol for multi-server architecture, Journal of Network and Computer Applications, Vol. 34, No. 2, **(2011),** pp. 609-618.
[11] W. J. Tsaur, J. H. Li, and W. B. Lee, An efficient and secure multi-server authentication scheme with key agreement, The Journal of System and Software, Vol. 85, **(2012),** pp. 876-882.
[12] C. Xu, Z. Jia, F. Wen, and Y. Ma, A novel of dynamic identity based authentication scheme for multi-server environment using smart cards, International Journal of Security and Its Applications, Vol. 7, No. 4, **(2013),** pp. 105-118.
[13] T. Jain, S. P. Singh, An Efficient and Secure Multi-server Smart Card based Authentication Scheme, International Journal of Computer Applications, Vol. 93, No. 12, **(2014),** pp. 1-7.

Author

Kwang Cheul Shin Division of Industrial Management Engineering, Sungkyul University. Sungkyul University-Ro 53, Anyang 8 dong, Manan-gu, Anyang-si, Gyeonggi-do, 14097, Korea. skcskc12@sungkyul.edu.

Education & Work experience: 2003, Ph.D. degree in Information and Communication Engineering, Sungkyunkwan University. Currently : Professor in Dept. of Industrial Management Engineering, Sungkyul University. Tel: 82-031-467-8916.