

# Cryptanalysis of Multiple-Server Password-Authenticated Key Agreement Schemes Using Smart Cards

Sang-Gon Lee, *Member, KIMICS*

**Abstract**— Password-based user-authentication schemes have been widely used when users access a server to avail internet services. Multiserver password-authentication schemes enable remote users to obtain service from multiple servers without separately registering with each server. In 2008, Jia-Lun Tsai proposed an improved and efficient password-authenticated key agreement scheme for a multiserver architecture based on Chang-Lee's scheme proposed in 2004. However, we found that Tsai's scheme does not provide forward secrecy and is weak to insider impersonation and denial of service attacks. In this article, we describe the drawbacks of Tsai's scheme and provide a countermeasure to satisfy the forward secrecy property.

**Index Terms** Multiserver Password-authentication, Insider Impersonation Attack, DoS Attack, Key Agreement.

## I. INTRODUCTION

**PASSWORD** authentication is one of the simplest and most convenient authentication mechanisms followed over public networks. Conventional password-authentication schemes are suitable for solving the privacy and security problems of a single user operating under a client-server architecture. As the network becomes larger, the password-authentication schemes that only support a single server are clearly not sufficient to address the users' growing needs. Therefore, many dynamic password-authentication schemes have been proposed for the multiserver architecture [1~5]. In the multiserver architecture, there are two types of servers—the service server and the control server. The service server is a public server that allows all legitimate users to access the network, whereas the control server is a private back end server, which is not disclosed. Multiserver password-authentication schemes enable remote users with a single password to access multiple servers without separately registering with each server. In addition, most of them provide an added functionality to derive a session key (SK), such as an encryption key or a message authentication code (MAC) key, for each session, particularly for subsequent secure communication use, to

protect the messages transmitted during each session. These schemes are therefore referred to as password-authenticated key agreement protocols.

In 2004, Juang proposed a password-authenticated key agreement scheme for a multiserver architecture using symmetric encryption methods without the need for a verification table [3]. However, Juang's scheme lacks efficiency, and each server needs to additionally protect and securely maintain an encrypted key table. In the same year, Chang and Lee proposed a similar but more efficient and secure scheme using symmetric encryption [1]. Their scheme is claimed to be able to achieve the following six vital requirements of the multiserver password-authentication scheme: choose and change password at will; less computation; security; mutual authentication; single registration; SK agreement. However Chang-Lee's scheme assumes that the registration centre (RC) and all the service servers are trustworthy and that the RC sends each server a shared secret key through a secure channel. This assumption is not practical because the service server is a public domain and thus is the major target of the adversary. Moreover, sharing of the secret key between the RC and the service server facilitates impersonation of each other. In 2007, Tsai proposed a password-authenticated key agreement scheme for a multiserver architecture using hash functions without the need for a verification table [6]. Tsai's scheme assumes that only the RC is trustworthy. This assumption is more practical because the service server is open to all legitimate users, but the control server is maintained private in the back end. However, we find that Tsai's scheme does not provide for forward secrecy under its assumption, and some potential threats to the scheme have been negligently unexplored in the corresponding security analysis. In this study, we will analyze these issues in detail.

The structure of this article is organized as follows. In the next section, we will recapitulate Tsai's multiserver password-authenticated key agreement scheme using smart cards. In section 3, we will discuss a few cryptographic flaws of Tsai's schemes and provide a countermeasure to satisfy the forward secrecy property. In the last section, we will conclude this article.

Manuscript received May 28, 2011; revised June 27, 2011; accepted July 8, 2011.

Sang-Gon Lee is with the Division of Computer & Information Engineering, Dongseo University, Busan, 617-716, Korea (Email: nok60@dongseo.ac.kr)

## II. REVIEW OF TSAI'S SCHEME

The main notations used throughout this article are listed as follows.

Symbol	Definition
RC	Registration centre.
$U_i$	User $U_i$ .
$S_j$	Service server $S_j$ .
$ID_{U_i}, ID_{S_j}$	Identity of user $U_i$ and service server $S_j$ , respectively.
$PW_{U_i}$	Password for user $U_i$ .
$N_{U_i}, N_{S_j}, N_{RC}$	Nonce generated by user $U_i$ , service server $S_j$ , and registration centre respectively.
$N_{S_j U_i}$	Nonce generated by $S_j$ for mutual authentication to $U_i$ .
$x$	Secret key of RC for users authentication.
$y$	Secret key of RC for servers authentication.
$h(\cdot)$	One-way hash function.
$\oplus$	XOR operation.
$\parallel$	Concatenation.
$\Rightarrow$	Secure channel transfer.
$\rightarrow$	Common channel transfer.

This subsection reviews Tsai's multiserver password-authenticated key agreement scheme [2]. Tsai's scheme consists of 4 phases: registration phase, login phase, authentication-server and RC phase, and authentication-server and user phase. The details of these phases can be described as follows.

### A. Registration phase

**Step R1.**  $U_i \Rightarrow RC : \{ID_{U_i}, PW_{U_i}\}$

Initially, a user  $U_i$ , with identity  $ID_{U_i}$ , chooses a password  $PW_{U_i}$  and submits  $ID_{U_i}, PW_{U_i}$  to the RC through a secure channel.

**Step R2.**  $RC \Rightarrow U_i$ : Smart cards with information  $\{h(\cdot), C_0\}$ .

On reception of the same, RC computes

$$R_{U_i} = h(ID_{U_i} \parallel x), \text{ and} \quad (1)$$

$$C_0 = R_{U_i} \oplus h(PW_{U_i}). \quad (2)$$

Then, the RC successfully issues a smart cards containing information  $\{h(\cdot), C_0\}$  to  $U_i$  through a secure channel.

### B. Login phase

**Step L1.**  $U_i$  inserts the smart card into a card reader and enters his/her identity  $ID_{U_i}$  and password  $PW_{U_i}$  to

request access to the server  $S_j$ .

**Step L2.**  $U_i \rightarrow S_j : ID_{U_i}, C_1$

The smart card extracts information  $h(ID_{U_i} \parallel x)$  from the smart card by computing  $h(ID_{U_i} \parallel x) = C_0 \oplus h(PW_{U_i})$ .

The reader generates a random nonce  $N_{U_i}$  and computes

$$C_1 = h(ID_{U_i} \parallel x) \oplus N_{U_i}. \quad (3)$$

The reader sends  $ID_{U_i}$  and  $C_1$  to the server  $S_j$ .

### C. Authentication-server and RC phase

**Step SR1.**  $S_j \rightarrow RC : ID_{U_i}, ID_{S_j}, C_1, C_2$

$S_j$  generates a random nonce  $N_{S_j}$ , and computes

$$C_2 = h(ID_{S_j} \parallel y) \oplus N_{S_j}. \quad (4)$$

Then  $S_j$  sends  $ID_{U_i}, ID_{S_j}, C_1$ , and  $C_2$  to the RC.

**Step SR2.**  $RC \rightarrow S_j : C_3$

The RC computes  $N'_{S_j} = C_2 \oplus h(ID_{S_j} \parallel y)$  and generates a random nonce  $N_{RC}$  to compute

$$C_3 = h(ID_{S_j} \parallel y) \oplus N_{RC}; \quad (5)$$

$C_3$  is then sent to  $S_j$ .

**Step SR3.**  $S_j \rightarrow RC : C_4$

$S_j$  retrieves  $N'_{RC}$  by computing  $C_3 \oplus h(ID_{S_j} \parallel y)$  and further computes

$$C_4 = h(h(ID_{S_j} \parallel y) \parallel N_{S_j}) \oplus N'_{RC}. \quad (6)$$

It sends  $C_4$  to the RC.

**Step SR4.**  $C'_4 \stackrel{?}{=} C_4$

The server  $S_j$  computes  $C'_4 = h(h(ID_{S_j} \parallel y) \parallel N'_{S_j}) \oplus N_{RC}$  and checks whether

$$C'_4 \stackrel{?}{=} C_4. \quad (7)$$

If Eq. (7) does not hold,  $S_j$  is legal, and the following steps are processed. If they are not equal,  $S_j$  is not legal, and RC terminates the session.

**Step SR5.**  $RC \rightarrow S_j : C_5, C_6$

RC retrieves information  $N'_{U_i} = h(ID_{U_i} \parallel x) \oplus C_1$  and next computes  $C_5$  and  $C_6$  as follows:

$$C_5 = h(h(ID_{S_j} \parallel y) \parallel N'_{S_j} \parallel N_{RC}), \text{ and} \quad (8)$$

$$C_6 = h(h(ID_{S_j} \parallel y) \parallel N'_{S_j} + 1 \parallel N_{RC} + 2) \oplus h(h(ID_{U_i} \parallel x) \parallel N'_{U_i}). \quad (9)$$

It sends  $C_5$  and  $C_6$  to  $S_j$ .

**Step SR6.**  $C'_5 \stackrel{?}{=} C_5$

$S_j$  computes  $C'_5 = h(h(ID_{S_j} \parallel y) \parallel N_{S_j} \parallel N'_{RC})$  and verifies whether

$$C'_5 \stackrel{?}{=} C_5. \quad (10)$$

If Eq. (10) does not hold,  $S_j$  terminates the session. If yes,  $S_j$  proceeds to the next phase.

#### D. Authentication-server and user phase

**Step SU1.**  $S_j \rightarrow U_i: V_2, C_9$

After successfully verifying the identity of the RC,  $S_j$  must obtain the user -authentication key.  $S_j$  computes  $C_7$  as follows;

$$\begin{aligned} C_7 &= C_6 \oplus h(h(ID_{S_j} \| y) \| N_{S_j} + 1 \| N'_{RC} + 2) \\ &= h(h(ID_{U_i} \| x) \| N'_{U_i}); \end{aligned} \quad (11)$$

and generates a random nonce  $N_{S_j U_i}$ . It also computes  $C_8, C_9$  and  $V_2$  as shown below;

$$C_8 = C_7 \oplus C_1 = C_7 \oplus h(ID_{U_i} \| x) \oplus N_{U_i}, \quad (12)$$

$$V_2 = C_7 \oplus N_{S_j U_i}, \quad \text{and}$$

$$(13)$$

$$C_9 = h(C_7 \| N_{S_j U_i}) \oplus C_8. \quad (14)$$

It sends  $V_2$  and  $C_9$  to  $U_i$

**Step SU2.**  $C'_9 \stackrel{?}{=} C_9$

$U_i$  computes  $C'_7 = h(h(ID_{U_i} \| x) \| N_{U_i}), N'_{S_j U_i} = C'_7 \oplus V_2$  and  $C'_8 = C'_7 \oplus C_1 = C'_7 \oplus h(ID_{U_i} \| x) \oplus N_{U_i}$ .

It also computes

$$C'_9 = h(C'_7 \| N_{S_j U_i}) \oplus C'_8, \quad (15)$$

and verifies whether

$$C'_9 \stackrel{?}{=} C_9. \quad (16)$$

If the result is negative,  $U_i$  terminates the session. If positive,  $U_i$  proceeds to the next step.

**Step SU3.**  $U_i \rightarrow S_j: C_{10}$

$U_i$  computes

$$C_{10} = h(C'_7 \| C'_8 \| N'_{S_j U_i}) \quad (17)$$

and sends it to  $S_j$ .

**Step SU4.**  $C'_{10} \stackrel{?}{=} C_{10}$

$S_j$  computes

$$C'_{10} = h(C_7 \| C_8 \| N_{S_j U_i}) \quad (18)$$

and verifies whether

$$C'_{10} \stackrel{?}{=} C_{10}. \quad (19)$$

If it does not hold,  $S_j$  aborts. If it holds,  $S_j$  computes the SK

$$SK = h(C_7 + 1 \| C_8 + 2 \| N_{S_j U_i} + 3). \quad (20)$$

$U_i$  also can compute the SK

$$SK = h(C'_7 + 1 \| C'_8 + 2 \| N'_{S_j U_i} + 3). \quad (21)$$

The SK is applied for the encryption of all communications between  $S_j$  and  $U_i$ .

### III. WEAKNESSES

#### A. Weakness to Forward Secrecy

Provision of the forward-secrecy property by a password-based authenticated key agreement protocol is

of the utmost importance for avoiding past SKs from being recovered by the compromise of any participating entity's long-term secret key. Tsai's scheme assumes that only the RC is trustworthy. The service server, which is open to all legitimate users, can be penetrated through various means, such as hackers, viruses, worms, misconfigurations, and disgruntled system administrators. Therefore, assuming that the adversary can obtain the service server's long-term authentication key,  $h(ID_{S_j} \| y)$ , is entirely reasonable.

Let us assume that an adversary has acquired a set of transaction messages of the previous session and the service server's long-term authentication key  $h(ID_{S_j} \| y)$ .

The adversary can compute  $N_{S_j}$ ,  $N_{RC}$ ,  $C_7$ , and  $N_{S_j U_i}$  using Eqs. (4), (5), (9) and (13), respectively, and  $C_8$  using Eqs. (3) and (11). Now, the adversary can compute the SK of the previous session with the knowledge obtained from the derived information  $\{C_7, C_8, N_{S_j U_i}\}$ .

**Countermeasure** From our experience, we can say that forward secrecy in a key agreement protocol cannot be achieved using only the hash function and symmetric encryption schemes. We need to use public-key techniques (e.g., exponentiations in a multiplicative group).

An efficient approach to the preservation of the forward-secrecy property is to slightly modify the protocol specification. Instead of choosing  $N_{U_i}$ ,  $N_{S_j}$ , and  $N_{RC}$  as random nonces, we redefine them as the user's, the server's, and the RC's ephemeral public key, computed respectively as follows:

$$N_{U_i} = g^{r_{U_i}}, \text{ for } r_{U_i} \in {}_R Z_q^*; \quad (22)$$

$$N_{S_j} = g^{r_{S_j}}, \text{ for } r_{S_j} \in {}_R Z_q^*; \quad (23)$$

and

$$N_{RC} = g^{r_{RC}}, \text{ for } r_{RC} \in {}_R Z_q^*. \quad (24)$$

Moreover, we include additional components  $N_{RC}^{r_{S_j}} = g^{r_{RC} r_{S_j}}$ ,  $N_{RC}^{r_{U_i}} = g^{r_{RC} r_{U_i}}$ , and  $(N_{RC}^{r_{U_i}})^{r_{S_j}} = (N_{RC}^{r_{S_j}})^{r_{U_i}} = (N_{RC}^{r_{S_j}})^{r_{U_i}} = g^{r_{RC} r_{U_i} r_{S_j}}$  in the SK derivation function such that

$$U_i : SK_{U_i S_j} = h((N_{RC}^{r_{U_i}})^{r_{S_j}} \| N_{RC}^{r_{S_j}} \| N_{RC}^{r_{U_i}} \| ID_{U_i} \| ID_{S_j} \| N_{S_j U_i}); \quad (25)$$

and

$$S_j : SK_{S_j U_i} = h((N_{RC}^{r_{S_j}})^{r_{U_i}} \| N_{RC}^{r_{U_i}} \| N_{RC}^{r_{S_j}} \| ID_{U_i} \| ID_{S_j} \| N_{S_j U_i}) \quad (26)$$

With these modifications, if an adversary happens to learn  $h(S_j \| y)$  after completion of a key-establishment session, he/she will only be able to derive  $N_{RC}^{r_{S_j}}$  and  $N_{RC}^{r_{U_i}}$  but not the shared secret  $g^{r_{U_i} r_{S_j} r_{RC}}$  (bound by the

intractability of the Computational Diffie-Hellman Problem) because he/she does not possess the knowledge of any private ephemeral key  $r_{U_i}$ ,  $r_{S_j}$ , or  $r_{RC}$ . Intuitively, the secrecy of the  $SK$  and the security of the scheme can be significantly preserved.

#### B. Insider impersonation attack

Tsai's scheme does not check the freshness of the nonce. Exploiting this fault, an adversary can impersonate as a legal user. Let us assume that an adversary has acquired a service server's long-term authentication key  $h(ID_{S_j} || y)$ . As the first step of the attack, the adversary replays the login message  $\{ID_{U_i}, C_1\}$  of  $U_i$  from the previous session to login. Because the adversary has the service server's long-term authentication key, he/she can pass the authentication-server and RC phase and generate a legal response  $C_{10}$  in the authentication-server and user phase. Finally, the adversary agrees to a new  $SK$  represented as follows:  $SK = h(C_7 + 1 || C_8 + 2 || N_{S_j U_i} + 3)$ .

Hence, we can easily deduce that Tsai's scheme cannot withstand Insider impersonation attack also.

#### C. Denial of service attack

In the specification for Tsai's protocol, the server and RC do not check whether the login request is legal until the session reaches the Step SU3 of the authentication-server and user phase. The adversary can launch denial of service (DoS) attacks by just sending login messages  $\{ID_{U_i}, C_{1A}\}$ , where  $C_{1A}$  is an arbitrary nonce chosen by the adversary. The adversary can deprive both the service server and the RC of computing power without incurring any cost except that of sending a message.

## IV. CONCLUSION

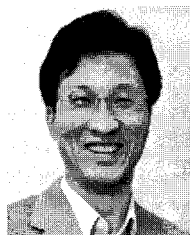
We have recapitulated the security features of Tsai's password-authenticated key agreement scheme. Particularly, we have pointed out the absence of the forward secrecy property and provided a countermeasure to satisfy the forward-secrecy property. In addition, we have further highlighted the threats possible due to KCI and DoS attacks. We hope that our discussion will provide adequate awareness to protocol designers for contemplating appropriate security considerations while designing future password-authenticated key agreement schemes to preclude the above-discussed vulnerabilities.

## ACKNOWLEDGMENT

This work was supported by 2011 National Research Foundation of Korea.

## REFERENCES

- [1] C.-C. Chang, J.-S. Lee, "An Efficient and Secure Multi-server Password Authentication Scheme using Smart Cards," *International Conference on Cyberworlds (CW '04)*, pp. 417-422, 2004.
- [2] L. Hu, X. Niu, Y. Yang, "An Efficient Multi-server Password Authenticated Key Agreement Scheme using Smart Cards," *International Conference on Multimedia and Ubiquitous Engineering (MUE '07)*, IEEE, pp.903-907, 2007.
- [3] W.-S. Juang, "Efficient Multi-server Password Authenticated Key Agreement using Smart Cards," *IEEE Trans. on Consumer Electronics*, vol. 50, no.1, pp. 251-255, 2004.
- [4] I.-C. Lin, M.-S. Hwang, and L.-H. Li, "New Remote User Authentication Scheme for Multi-server Architecture," *Future Generation Computer Systems*, vol. 19, pp. 13-22, 2003.
- [5] L.-H. Li, I.-C. Lin and M.-S. Hwang, "A Remote Password Authentication Scheme for Multi-server Architecture using Neural Networks," *IEEE Trans. on Neural Networks*, vol. 12, pp.1498-1504, 2001.
- [6] J.-L. Tsai, "Efficient multi-server authentication schemes based on one-way hash function verification table," *Computers & Security*, vol. 27, no.3-4, pp. 115-121, 2008.



**Sang-Gon Lee** received the BEng, MEng, and PhD degree in electronics engineering from Kyungpook National University, Korea, in 1986, 1988, and 1993, respectively. He is a professor at the Division of Computer & Information Engineering, Dongseo University. He was an assistant/associate professor at Chang-shin College from 1991 to 1997 and a visiting scholar at QUT, Australia from 2003 to 2004. His research interest areas include

wireless mesh network, future internet, information security, network security.