Comments of an efficient and secure multi-server authentication scheme with key agreement

Yitao Chen

School of Mathematics and Statistics, Wuhan University, Wuhan, People's Republic of China

Email: chenyitao.math@gmail.com

Abstract*:* Recently, Tsaur et al. proposed an authentication scheme for multi-server environments and claimed their scheme could withstand various attacks. In this letter, we will point out that Tsaur et al. scheme is not suitable for multi-server environments since the user has to register for every server. Furthermore, we will show Tsaur et al. scheme is vulnerable to the password guessing attack and the privileged insider attack.

Key words*: multi-server, password authentication protocol, smart card, password change, key agreement*

1. Introduction

The authentication protocols using smart cards are widely used in the network communication. To satisfy the application in multi-server environments, the authentication scheme for multi-server environments was proposed. Recently, Tsaur et al. [1] proposed an authentication scheme for multi-server environments and claimed their scheme could withstand various attacks. In this paper we will show Tsaur et al.'s scheme is not suitable for multi-server environments. We also show Tsaur et al.'s scheme is vulnerable to the password guessing attack and the privileged insider attack.

The organization of the letter is sketched as follows. The Section 2 gives a brief review of Tsaur et al.'s scheme. The weanesses of Tsaur et al.'s scheme are shown in Section 3. Finally, we give some conclusions in Section 4.

2. Tsaur et al.'s scheme

In this section, we will briefly review Tsaur et al.'s scheme. As shown in Fig. 1.[1], their scheme consists of two phases: Registration phase and log-in and session key agreement phase. In order to facilitate future references, frequently used notations are listed below with their descriptions.

- $E_k(\cdot)$: The encryption function with secret key k;
- $D_k(\cdot)$: The decryption function with secret key k;
- : The bitwise exclusive-or operator;
- ||: The concatenation operator;
- $h(\cdot)$: A one-way and collision-free hash function;
- RC: The registration center;
- S_j: The j th server;
- U_i: The i th user;
- x: The secret key of the registration center;
- SID_j: The j th server's identity;
- UID_i: The i th user's identity;
- w_j: The secret key shared between RC and S_j;
- PW_i: The i th user's password;
- E_T_{ij}: The service period of S_j for U_i;
- v_i, u_i: U_i's secret information;
- v_{ij}: The secret key shared between U_i and S_j;
- A_{ij}: The authentication parameter for U_i to log in S_j;
- ru_k: A k th random value chosen by the smart card;
- M_{ij}: An authentication message for U_i to log in S_j;
- rs_k: The k th random value chosen by S_j;
- sk_k: The k th session key;
- T: A timestamp;

2.1. Registration phase

Suppose that user U_i want to get service granted from the server S_j. U_i first chooses his/her identity UID_i and password PW_i, and then sends them to RC via a secure channel. RC will perform the following steps:

1) RC computes $v_i = h(x+1, UID_i)$, $u_i = v_i \oplus h(PW_i)$.

2) RC computes $v_{ij} = h(v_i, SID_j)$ and $A_{ij} = E_{w_i \oplus E\_T_{ij}}(v_{ij})$.

3) RC stores UID_i, u_i, E_T_{ij} and A_{ij} to the memory of a smart card and issue this smart card to U_i.

2.2. Password authentication phase

Once the client A wants to login to the server S, he will perform the following login steps.

1) The user U_i inputs his identity password PW_i into the terminal. The smart card generates a random number ru_k, computes $v_i = u_i \oplus h(PW_i)$, $v_{ij} = h(v_i, SID_j)$ and $E_{v_{ij}}(ru_k, h(UID_i))$. Then U_i sends the message $M_1 = \{E\_T_{ij}, A_{ij}, UID_i, E_{v_{ij}}(ru_k, h(UID_i))\}$ to S_j.

2) Upon receiving the message M_1, S_j computes $v_{ij} = D_{w_i \oplus E\_T_{ij}}(A_{ij})$, decrypts $E_{v_{ij}}(ru_k, h(UID_i))$ and verifies the correctness of $h(UID_i)$. Then S_j generates a random number rs_k, chooses a timestamp T and computes the session key $sk_k = (rs_k, ru_k, v_{ij})$. At last, S_j sends $M_2 = \{E_{v_{ij}}(rs_k, ru_k, T)\}$.

3) Upon receiving the message M_2, U_i decrypts $E_{v_{ij}}(rs_k, ru_k, T)$ and checking the correctness of ru_k. If ru_k is correct, U_i computes the session key $sk_k = (rs_k, ru_k, v_{ij})$ and sends $M_3 = \{E_{sk_k}(T, sk_k)\}$ to S_j.

4) Upon receiving the message M_3, S_j decrypts $E_{sk_k}(T, sk_k)$ using sk_k. The S_j checks whether the freshness of T by checking whether $t_{new} - T > \Delta T$. If $t_{new} - T > \Delta T$, S_j stops the session. Otherwise, U_i is authenticated.

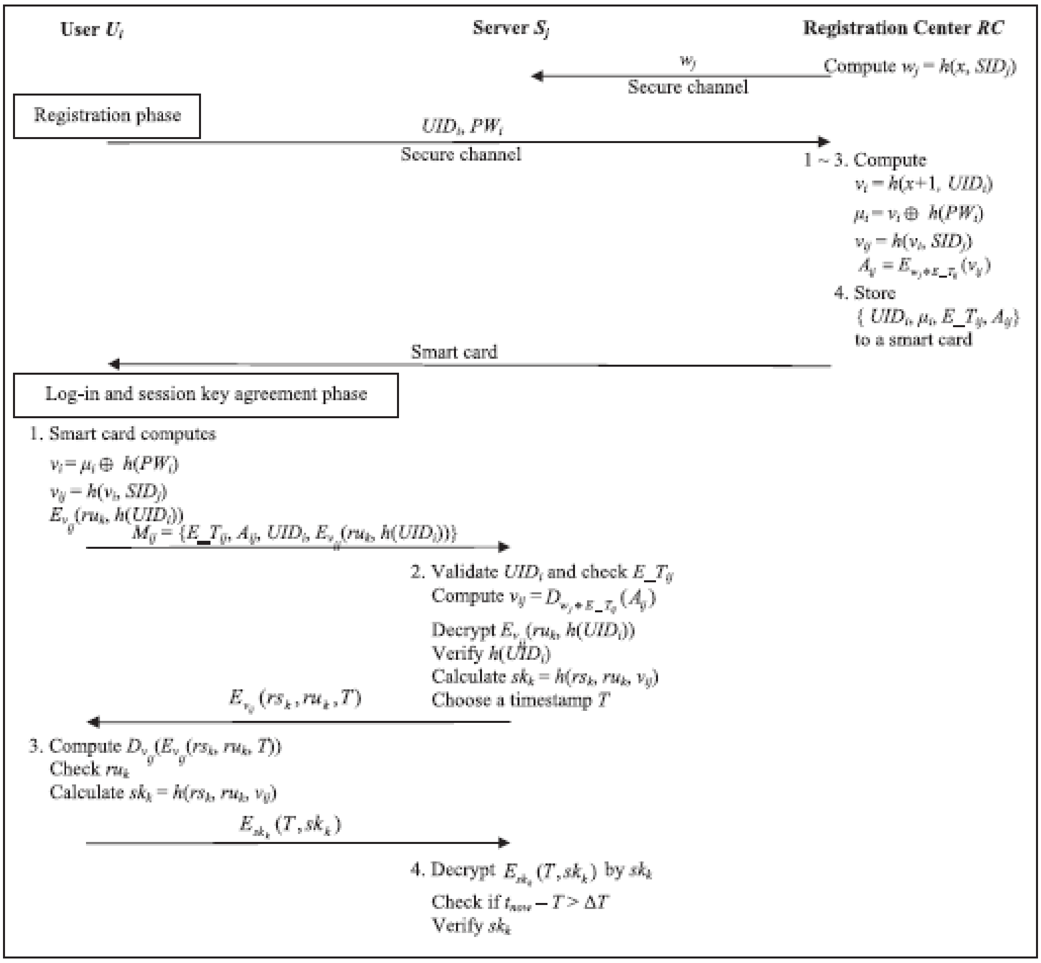
User U_i
Server S_j
Registration Center RC
w_j
Compute $w_j = h(x, SID_j)$
Secure channel
Registration phase
UID_i, PW_i
Secure channel
1 ~ 3. Compute
$v_i = h(x+1, UID_i)$
$\mu_i = v_i \oplus h(PW_i)$
$v_{ij} = h(v_i, SID_j)$
$A_{ij} = E_{w_j+E\_T_{ij}}(v_{ij})$
4. Store
$\{UID_i, \mu_i, E\_T_{ij}, A_{ij}\}$
to a smart card
Smart card
Log-in and session key agreement phase
1. Smart card computes
$v_i = \mu_i \oplus h(PW_i)$
$v_{ij} = h(v_i, SID_j)$
$E_{v_{ij}}(ru_k, h(UID_i))$
$M_{ij} = \{E\_T_{ij}, A_{ij}, UID_i, E_{v_{ij}}(ru_k, h(UID_i))\}$
2. Validate UID_i and check E_T_{ij}
Compute $v_{ij} = D_{w_j+E\_T_{ij}}(A_{ij})$
Decrypt $E_{v_{ij}}(ru_k, h(UID_i))$
Verify $h(UID_i)$
Calculate $sk_k = h(rs_k, ru_k, v_{ij})$
Choose a timestamp T
$E_{v_{ij}}(rs_k, ru_k, T)$
3. Compute $D_{v_{ij}}(E_{v_{ij}}(rs_k, ru_k, T))$
Check ru_k
Calculate $sk_k = h(rs_k, ru_k, v_{ij})$
$E_{sk_k}(T, sk_k)$
4. Decrypt $E_{sk_k}(T, sk_k)$ by sk_k
Check if $t_{now} - T > \Delta T$
Verify sk_k

Fig. 1. Work flow of Tsaur et al.'s scheme

3. Weaknesses of Tsaur et al.'s scheme

In this section, we will show Tsaur et al.'s scheme is not suitable for the multi-server environment since the user has to register to every server. We also show Tsaur et al.'s scheme is vulnerable to the password guessing attack and the privileged Insider attack.

3.1 No single registration

Many requirements have been proposed for the authentication scheme for multi-server environments. Single registration is the fundamental requirement in all the requirements, i.e. any user only must register at the registration centre once and can use all the permitted services in remote servers. From the description in Section 2, we know the users must get service granted from multiple servers with repeating registration to each server. This is not user friendly. Then Tsaur et al.'s scheme is not for multi-server environments.

3.2. Password guessing attack

Kocher et al. [2] and Messerges et al. [3] have pointed out that all existent smart cards are vulnerable in that the confidential information stored in the device could be extracted by physically monitoring its power consumption; once a card is lost, all secrets in it may be revealed. To evaluate the security of smart card based user authentication, we assume the capabilities that an adversary $\mathscr{A}$ may have as follows:

1) The adversary has total control over the communication channel between the users and the server in the login and authentication phases. That is, $\mathscr{A}$ may intercept, insert, delete, or modify any message in the channel.

2) $\mathscr{A}$ may (i) either steal a user's smart card and then extract the information from it, (ii) or obtain a user's password, (iii) but not both (i) and (ii).

Suppose an adversary $\mathscr{A}$ has stolen U_i's smart card and extracted the stored values UID_i , u_i , E_T_{ij} and A_{ij} , where $v_i = h(x+1, UID_i)$, $u_i = v_i \oplus h(PW_i)$, $v_{ij} = h(v_i, SID_j)$ and $A_{ij} = E_{w_i \oplus E\_T_{ij}}(v_{ij})$. Then the attacker $\mathscr{A}$ could impersonate U_i to login in the server by performing the following procedure.

1) $\mathscr{A}$ collects a message $M_1 = \{E\_T_{ij}, A_{ij}, UID_i, E_{v_{ij}}(ru_k, h(UID_i))\}$ transmitted between U_i and S_j .

2) $\mathscr{A}$ guesses a password PW_i' , computes $v_i' = v_i \oplus h(PW_i')$ and $v_{ij}' = h(v_i', SID_j)$.

3) $\mathscr{A}$ gets ru_k' and $h(UID_i)'$ by decrypting $E_{v_{ij}}(ru_k, h(UID_i))$.

4) $\mathscr{A}$ checks whether $h(UID_i)'$ and $h(UID_i)$ are equal. If they are equal, $\mathscr{A}$ finds the correct password. Otherwise, $\mathscr{A}$ repeats 1)-4) until finding the correct password.

From the above description, we know the adversary can get the password. Therefore, Tsaur et al.'s scheme is vulnerable to the password guessing attack.

3.3. Privileged Insider attack

In a real environment, it is a common practice that many users use same passwords to access different applications or servers for their convenience of remembering long passwords and ease-of-use whenever required [4]. However, if the system manager or a privileged insider $\mathscr{A}$ of the register centre RC knows the passwords of user U_i, he may try to impersonate U_i by accessing other servers where U_i could be a registered user. In the user registration phase of Tsaur et al.'s scheme, U_i sends the password PW_A to RC. Then, the privileged-insider of RC could get the password easily. Therefore, Tsaur et al.'s scheme is vulnerable to the privileged insider attack.

4. Conclusion

Recently, Tsaur et al. proposed an authentication scheme for multi-server environments and demonstrated its immunity against various attacks. However, after review of their scheme and analysis of its security, three kinds of weaknesses are presented in different scenarios. The analyses show that the scheme is insecure for practical application.

Reference

[1]. Tsaur, W.-J., et al., An efficient and secure multi-server authentication scheme with key agreement. J. Syst. Software (2011), doi:10.1016/j.jss.2011.10.049

[2]. P. Kocher, J. Jaffe, B. Jun, Differential power analysis, Proc. Advances in Cryptology (CRYPTO'99), (1999) 388 – 397.

[3]. T.S. Messerges, E.A. Dabbish, R.H. Sloan, Examining smart card security under the threat of power analysis attacks, IEEE Transactions on Computers 51 (5) (2002) 541–552.

[4]. H.C.Hsiang, W.K. Shiha, Improvement of the secure dynamic ID based remote user authentication next term scheme for multi-server environment, Computer Standards & Interfaces, 31(6) (2009) 1118-1123.