Cryptanalysis of Hsiang-Shih's Secure Dynamic ID Based Remote User Authentication Scheme

Wen-Chung Kuo
Department of Computer Science and Information EngineerinG
National Formosa University
simonkuo@nfu.edu.tw

Yu-Shuan Chu
Department of Computer Science and Information Engineering
National Formosa University
s49343101@nfu.edu.tw

Bae-Ling Chen
Graduate School of Engineering Science and Technology
National Yunlin University of Science and Technology
chenbl@yuntech.edu.tw

***Abstract*— Recently, Liao and Wang proposed a secure dynamic ID based remote user authentication scheme for multi-server environment. They achieved user's anonymity by using secure dynamic ID instead of static ID. Later, Hsiang and Shih gave an improved scheme to repair the security flaws found in Liao-Wang's scheme. Hsiang and Shih claimed that their scheme inherits the merits, enhances the security of Liao-Wang's scheme, and achieves mutual authentication that Liao-Wang's scheme fails to provide. In this paper, however, we show that Hsiang-Shih's scheme cannot withstand both user and server impersonation attacks. In addition, their scheme is vulnerable to malicious user and insecure for practical application.**

***Index Terms*—Cryptanalysis, Authentication, Smart Card, Dynamic ID, Multi-server.**

I. INTRODUCTION

With the increasing number of systems that provide services over open networks, remote authentication is critical for preventing unauthorized parties from accessing remote system resources. Smart card based authentication schemes are the most commonly used mechanism in remote user authentication schemes. With the convenience of networks, the system resources or services are often composed of many different servers distributed over the network to make remote users access efficiently and conveniently. Most of traditional authentication schemes use real ID or static ID for multi-server environment, but this careless design causes that adversary is able to trace and identify user(s) requests by monitoring the communications between servers [3].

Recently, Liao and Wang proposed a secure dynamic ID based remote user authentication scheme for multi-server environment [3]. Their scheme uses only hashing functions in mutual authentication and session key agreement and dynamic ID [1] instead real ID or static ID to achieve user's anonymity. They claimed their scheme can get service granted from multi-server environment. Later, Hsiang and Shih gave an improved scheme [2] to repair the security flaws found in Liao-Wang's scheme. Hsiang and Shih claimed that their scheme inherits the merits, enhances the security of Liao-Wang's scheme, and achieves mutual authentication that Liao-Wang's scheme fails to provide. In this paper, however, we show that Hsiang-Shih's scheme cannot withstand both user and server impersonation attacks. In addition, their scheme is vulnerable to malicious user and insecure for practical application.

The rest of this paper is organized as follows. In Section 2, we briefly review Hsiang-Shih's scheme. We analyze the weakness of Hsiang-Shih's scheme in Section 3. Our conclusions are given in Section 4.

II. REVIEW OF HSIANG-SHIH'S SCHEME

In this section, we briefly review Hsiang-Shih's scheme. For convenience, the notations used in Hsiang-Shih's scheme are listed as follows:

● RC	registration center
● x	master secret key of RC
● r, y	secret numbers of RC
● U_i	i-th user
● ID_i	identification of U_i

CID_i	dynamic ID of U_i
pw_i	password of U_i
b_i	blind factor of U_i
S_j	j-th remote server
SID_j	identification of S_j
$h(\cdot)$	secure one-way hash
$\oplus$	bitwise XOR operation
$\|\|$	string concatenation operation

Hsiang-Shih's scheme assumes that only *RC* knows the master secret key x and two secret numbers r, y. There are four phases in Hsiang-Shih's scheme: the registration phase, the login phase, the mutual authentication and session key agreement phase, and the password change phase.

A. Registration phase

In the registration phase, user U_i initially registers with registration center *RC*. U_i submit his identity ID_i and password pw_i to registration center *RC*, and *RC* performs the following steps:

Step R1. U_i chooses his password pw_i and arbitrary number b_i, and then, computes $h(b_i \oplus pw_i)$.

Step R2. U_i sends $\{ID_i, h(b_i \oplus pw_i)\}$ to *RC* over a secure channel.

Step R3. Upon receiving the registration information, *RC* performs following computations:

$$T_i = h(ID_i \| x)$$
$$V_i = T_i \oplus h(ID_i \| h(b_i \oplus pw_i))$$
$$A_i = h(h(b_i \oplus pw_i) \| r) \oplus h(x \oplus r)$$
$$B_i = A_i \oplus h(b_i \oplus pw_i)$$
$$R_i = h(h(b_i \oplus pw_i) \| r)$$
$$H_i = h(T_i).$$

Step R4. *RC* issues a smart card containing $\{V_i, B_i, H_i, R_i, h(\cdot)\}$ to U_i over a secure channel.

Step R5. Upon receiving the smart card, U_i enters b_i into his smart card.

Note that U_i's smart card contains $\{V_i, B_i, b_i, R_i, H_i, h(\cdot)\}$.

B. Login phase

This phase is invoked whenever U_i requests to log into S_j. U_i inputs his identity ID_i, password pw_i, and the identity of target server SID_j to his smart card, and the smart card performs the following steps:

Step L1. U_i's smart card computes $T_i = V_i \oplus h(ID_i \| h(b_i \oplus pw_i))$ and $H_i^* = h(T_i)$ and checks whether H_i^* and H_i is equal. If they are not equal, the smart card rejects U_i; otherwise, the legitimacy of U_i can be assured.

Step L2. The smart card generates nonce N_i and performs the following computations:

$$A_i = B_i \oplus h(b_i \oplus pw_i)$$
$$CID_i = h(b_i \oplus pw_i) \oplus h(T_i \| A_i \| N_i)$$
$$P_{ij} = T_i \oplus h(A_i \| N_i \| SID_j)$$
$$Q_i = h(B_i \| A_i \| N_i)$$
$$D_i = R_i \oplus SID_j \oplus N_i$$
$$C_0 = h(A_i \| N_i + 1 \| SID_j).$$

Step L3. The smart card sends U_i's login request $\{CID_i, P_{ij}, Q_i, D_i, C_0, N_i\}$ to S_j.

C. Mutual verification and session key agreement phase

In this phase, user U_i and server S_j authenticate each other. After finish mutual authentication protocol, U_i and S_j compute their session key *SK* respectively. U_i and S_j perform the following steps:

Step V1. Upon receiving the login request, S_j generates nonce N_{jr} and computes $M_{jr} = h(SID_j \| y) \oplus N_{jr}$, then sends the message $\{M_{jr}, SID_j, D_i, C_0, N_i\}$ to registration center *RC*.

Step V2a. Upon receiving S_j's message, *RC* computes $N_{jr}' = M_{jr} \oplus h(SID_j \| y)$, $R_i' = D_i \oplus SID_j \oplus N_i$, and $A_i' = R_i' \oplus h(x \oplus r)$.

Step V2b. *RC* computes $C_0' = h(A_i \| N_i + 1 \| SID_j)$ and compares it with C_0. If they are not equal, *RC* terminates the authentication protocol.

Step V2c. *RC* generates nonce N_{rj} and computes $C_1 = h(N_{jr}' \| h(SID_j \| y) \| N_{rj})$ and $C_2 = A_i \oplus h(h(SID_j \| y) \| N_{rj})$, and sends $\{C_1, C_2, N_{rj}\}$ back to S_j.

Step V3. Upon receiving *RC*'s reply, S_j computes $C_1' = h(N_{jr} \| h(SID_j \| y) \| N_{rj})$ compares it with C_1. If they are not equal, S_j reports a *RC* authentication error and terminates the authentication protocol.

Step V4. S_j computes $A_i = C_2 \oplus h(h(SID_j \| y) \| N_{rj})$, $T_i = P_{ij} \oplus h(A_i \| N_i \| SID_j)$, $h(b_i \oplus pw_i) = CID_i \oplus h(T_i \| A_i \| N_i)$, and $B_i = A_i \oplus h(b_i \oplus pw_i)$.

Step V5. S_j computes $h(B_i \| A_i \| N_i)$ and compares it with Q_i. If they are not equal, S_j terminates the authentication protocol.

Step V6. S_j generates nonce N_j, computes $M_{ji}' = h(B_i \| N_i \| A_i \| SID_j)$, and sends back $\{M_{ji}', N_j\}$ to U_i.

Step V7. Upon receiving S_j's reply, U_i computes $h(B_i \| N_i \| A_i \| SID_j)$ and compares it with M_{ji}'. If they are not equal, U_i aborts the connection; otherwise S_j is authenticated by U_i.

Step V8. U_i computes $M_{ij}'' = h(B_i \| N_j \| A_i \| SID_j)$ and sends back M_{ij}'' to S_j.

Step V9. Upon receiving U_i's reply, S_j computes $h(B_i \| N_j \| A_i \| SID_j)$ and compares it with

M_{ji}”. If they are not equal, S_j terminates the authentication protocol; otherwise U_i is authenticated by S_j and the mutual authentication is completed. U_i and S_j then compute $h(B_i \| A_i \| N_i \| N_j \| SID_j)$ as their session key SK.

D. Password change phase

In this phase, user U_i can update his password without the help of registration center RC. U_i and his smart card perform the following steps:

Step C1. U_i inserts his smart card to his card reader, inputs $\{ID_i, pw_i\}$, and requests to change password.

Step C2. Upon receiving U_i’s request, the smart card computes $T_i = V_i \oplus h(ID_i \| h(b_i \oplus pw_i))$ and $H_i^* = h(T_i)$ and checks whether H_i^* and H_i is equal. If they are not equal, the smart card rejects U_i; otherwise, U_i is asked to choose new password pw_{inew}.

Step C3. After U_i inputs pw_{inew}, U_i’s smart card computes $V_{inew} = T_i \oplus h(ID_i \| h(b_i \oplus pw_{inew}))$ and $B_{inew} = B_i \oplus h(b_i \oplus pw_i) \oplus h(b_i \oplus pw_{inew})$. Finally, V_{inew} and B_{inew} are stored back to the smart card to replace V_i and B_i respectively.

III. WEAKNESS OF HSIANG-SHIH’S SCHEME

A. User impersonation attack

We first prove that a malicious user can easily impersonate other user without user’s password and smart card in Hsiang-Shih’s scheme. Suppose that there is a malicious user with identity U_a in Hsiang-Shih’s scheme. Since U_a is authenticated by remote server S_j, U_a has a smart card containing $\{V_a, B_a, b_a, R_a, H_a, h(\cdot)\}$, and these authentication information are known by U_a. U_a manipulates the authentication information stored on the smart card and the collected communication flows of another user U_i to impersonate U_i as the following steps:

Step U1. U_a first computes $A_a = B_a \oplus h(b_a \oplus pw_a)$, and then he has $h(x \oplus r) = R_a \oplus A_a$.

Step U2. From the collected communication flows of user U_i, U_a retrieves U_i’s login request $\{CID_i, P_{ij}, Q_i, D_i, C_0, N_i\}$ and performs the following computations:
$R_i = D_i \oplus SID_j \oplus N_i$
$A_i = R_i \oplus h(x \oplus r)$
$T_i = P_{ij} \oplus h(A_i \| N_i \| SID_j)$
$h(b_i \oplus pw_i) = CID_i \oplus h(T_i \| A_i \| N_i)$
$B_i = A_i \oplus h(b_i \oplus pw_i)$.

Step U3. U_a generates nonce N_a and performs the following computations:
$CID_i^* = h(b_i \oplus pw_i) \oplus h(T_i \| A_i \| N_a)$
$P_{ij}^* = T_i \oplus h(A_i \| N_a \| SID_j)$
$Q_i^* = h(B_i \| A_i \| N_a)$
$D_i^* = R_i \oplus SID_j \oplus N_a$
$C_0^* = h(A_i \| N_a + 1 \| SID_j)$.

Step U4. U_a sends the forged login request $\{CID_i^*, P_{ij}^*, Q_i^*, D_i^*, C_0^*, N_a\}$ to S_j.

Step U5. Upon receiving the login request, S_j generates nonce N_{jr} and computes $M_{jr} = h(SID_j \| y) \oplus N_{jr}$, then sends the message $\{M_{jr}, SID_j, D_i^*, C_0^*, N_a\}$ to registration center RC.

Step U6a. Upon receiving S_j’s message, RC computes $N_{jr}^* = M_{jr} \oplus h(SID_j \| y)$, $R_i^* = D_i^* \oplus SID_j \oplus N_a$, and $A_i^* = R_i^* \oplus h(x \oplus r)$.

Step U6b. RC computes $C_0^* = h(A_i^* \| N_a + 1 \| SID_j)$ and checks $C_0^* = C_0$.

Step U6c. RC generates nonce N_{rj} and computes $C_1^* = h(N_{jr}^* \| h(SID_j \| y) \| N_{rj})$ and $C_2^* = A_i^* \oplus h(h(SID_j \| y) \| N_{rj})$, and sends $\{C_1^*, C_2^*, N_{rj}\}$ back to S_j.

Step U7. Upon receiving RC’s reply, S_j computes $C_1’ = h(N_{jr} \| h(SID_j \| y) \| N_{rj})$ and checks $C_1’ = C_1^*$.

Step U8. S_j computes $A_i’ = C_2^* \oplus h(h(SID_j \| y) \| N_{rj})$, $T_i’ = P_{ij}^* \oplus h(A_i’ \| N_a \| SID_j)$, $h(b_i \oplus pw_i)’ = CID_i^* \oplus h(T_i’ \| A_i’ \| N_a)$, and $B_i’ = A_i’ \oplus h(b_i \oplus pw_i)’$.

Step U9. S_j computes $Q_i’ = h(B_i’ \| A_i’ \| N_a)$ and checks $Q_i’ = Q_i$.

Step U10. S_j generates nonce N_j, computes $M_{ji}^* = h(B_i’ \| N_a \| A_i’ \| SID_j)$, and sends back $\{M_{ji}^*, N_j\}$ to U_a.

Step U11. Upon receiving S_j’s reply, U_a computes $h(B_i \| N_a \| A_i \| SID_j)$ and checks it equals to M_{ji}^*.

Step U12. U_a computes $M_{ij}^{**} = h(B_i \| N_j \| A_i \| SID_j)$ and sends back M_{ij}^{**} to S_j.

Step U13. Upon receiving U_a’s reply, S_j computes $h(B_i’ \| N_j \| A_i’ \| SID_j)$ and checks it equals to M_{ji}^{**}. U_a is authenticated as U_i by S_j and the mutual authentication is completed. U_a can also compute $SK = h(B_i \| A_i \| N_i \| N_j \| SID_j)$.

The forged login request is accepted. S_j is fooled into believing that malicious user U_a is U_i. S_j authenticates U_a, and U_a access the remote system as U_i. Hence, U_a impersonate U_i without U_i’ password and smart card. Therefore, Hsiang-Shih’s scheme is vulnerable to user impersonation attacks.

B. Server impersonation attack

In this subsection, we show that a malicious user can easily impersonate remote server without the secret information sharing between servers and registration center in Hsiang-Shih's scheme. Suppose that there is a malicious user with identity U_a in Hsiang-Shih's scheme. U_a is trying to impersonate remote server S_j to cheat user U_i. U_i sends his login request $\{CID_i, P_{ij}, Q_i, D_i, C_0, N_i\}$ to U_a. By $h(x \oplus r) = R_a \oplus A_a$ and the same manner discussed in previous subsection (Section 3.1), U_a can get $R_i = D_i \oplus SID_j \oplus N_i$, $A_i = R_i \oplus h(x \oplus r)$, $T_i = P_{ij} \oplus h(A_i \| N_i \| SID_j)$, $h(b_i \oplus pw_i) = CID_i \oplus h(T_i \| A_i \| N_i)$, and $B_i = A_i \oplus h(b_i \oplus pw_i)$. Since U_a has A_i, T_i, $h(b_i \oplus pw_i)$, and B_i, U_a can compute $h(B_i \| A_i \| N_i)$ and check Q_i directly without the help of registration center RC. Beside, Ua can choose N_j, compute $M_{ji}' = h(B_i \| N_i \| A_i \| SID_j)$, and challenge U_i by message $\{ M_{ji}', N_j\}$.

U_i is fooled into believing that malicious user U_a is S_j. Hence, U_a impersonate S_j without the help of RC. Therefore, Hsiang-Shih's scheme is vulnerable to server impersonation attacks.

C. **Security flaw**

From the results of above two subsection, we know that in Hsiang-Shih's scheme, a legitimate user can easily compute $h(x \oplus r)$, so any legitimate user can execute impersonation attacks. Obviously, Hsiang-Shih's scheme fails to provide mutual authentication.

IV. CONCLUSIONS

In 2009, Hsiang and Shih proposed secure dynamic ID based remote user authentication scheme for multi-server environment. They claimed that their scheme inherits the merits and enhances the security of Liao-Wang's scheme, and achieves mutual authentication that Liao-Wang's scheme fails to provide. However, we have demonstrated that Hsiang-Shih's scheme suffers from both user and server impersonation attacks. In Hsiang-Shih's scheme, a malicious user can easily impersonate other user to access remote servers without correct password, and a malicious user can also impersonate any remote server to cheat other user without secret information of registration center. For this reason, their scheme is insecure for practical application.

REFERENCES

[1] M.L. Das, A. Saxena, and V.P. Gulati, "A dynamic ID-based remote user authentication scheme," IEEE Transactions on Consumer Electronics 50 (2) (2004) 629–631.

[2] 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 31 (6) (2009) 1118–1123.

[3] Y.P. Liao and S.S. Wang, "A secure dynamic ID based remote user authentication scheme for multi-server environment," Computer Standards & Interfaces 31 (1) (2009) 24–29.