Security Weaknesses of Two Dynamic ID-based User Authentication and Key Agreement Schemes for Multi-server Environment

Yun-Hsin Chuang
Department of Mathematics, National Changhua University of Education, Jin-De Campus, Chang-Hua 500, Taiwan
mathbaby@gmail.com

Yuh-Min Tseng
Department of Mathematics, National Changhua University of Education, Jin-De Campus, Chang-Hua 500, Taiwan
ymtseng@cc.ncue.edu.tw

***Abstract*— A remote user authentication scheme for multi-server environment provides mutual authentication and session key establishment between users and multiple servers. Recently, two dynamic ID-based remote user authentication schemes for multi-server environment were proposed. In this article, we analyze the security of both schemes. One scheme was proposed by Geng and Zhang. And we show that the proposed scheme suffers from a user-spoofing attack. In 2009, Hsiang and Shih also proposed the other scheme. We show that Hsiang and Shih's scheme is vulnerable to an insider attack and a server-spoofing server attack.**

***Index Terms*— security, user authentication, key agreement, multi-server, anonymous.**

I. INTRODUCTION

With the popularity of Internet, more and more applications are constructed on multi-server environment, in which users may access multiple servers remotely. In this multi-server environment, the system often consists of many different servers around the world, which provides services or resources to be accessed over open communication networks. For providing mutual authentication between users and servers, there are three kinds of approaches: password-based, public-key based and ID-based authentications.

Traditional remote user authentication is only suitable for solving the privacy and security problems in the single server architecture. The issue of remote login authentication for the single server environment has already been solved by a variety of schemes [3, 6, 9, 15]. If the traditional remote user authentication schemes are applied to the multi-server environment, each user must register and remember many credentials for multiple servers. Therefore, a secure remote user authentication scheme for the multi-server environment is needed to solve this problem. Several schemes [1, 2, 7, 11, 14] have been presented to study accessing the resources securely in the multi-server environment.

In some situations, users want to access the resources of the service providers anonymously. Several schemes [8, 10, 12] have been proposed to solve this issue. These schemes use dynamic IDs to login the service providers to achieve user's anonymity. However, these schemes are only suitable for the single server environment. Recently, to develop a dynamic ID-based user authentication scheme for the multi-server environment becomes a new research issue. In 2008, Geng and Zhang [4] proposed a dynamic ID-based user authentication and key agreement scheme for the multi-server environment using bilinear pairings. In 2009, Liao and Wang [13] also proposed a dynamic ID-based user authentication scheme for the multi-server environment. Later on, Hsiang and Shih showed that Liao and Wang's scheme is vulnerable to insider attack, masquerade attack, server-spoofing attack, and registration center spoofing attack. Meanwhile, Hsiang and Shih [5] also proposed an improvement on the Liao-Wang scheme to remedy these attacks.

In this paper, unfortunately, we will demonstrate the security weaknesses of two recently proposed schemes. We show that Hsiang and Shih's scheme [5] is vulnerable to an insider attack and a server-spoofing attack. For Geng and Zhang's scheme [4], we will show that their scheme suffers from a user-spoofing attack. The remainder of this paper is organized as follows. We review and show the security weaknesses of the Hsiang-Shih and the Geng-Zhang schemes in Section 2 and 3, respectively. Section 4 draws our conclusion and future

work.

II. Analysis of Hsiang and Shih's Scheme

In this section, we briefly review Hsiang and Shih's scheme and show their security weaknesses.

A. Review of Hsiang and Shih's scheme

Without loss of generality, suppose that the multi-server system consists of one registration center (*RC*), *m* users and *n* service providers. The notations used in this scheme are summarized as follows:

- $h(\ )$: a one-way hash function.
- S_j: the *j*-th server.
- U_i: the *i*-th user.
- ID_i: the identity of U_i.
- PW_i: the password of U_i.
- *RC*: the registration center.
- r, x, y : the secret keys of *RC*.
- SID_j: the identity of S_j.
- $\oplus$: the exclusive-or operation.
- $||$: the concatenation operation.

The registration center *RC* knows a master secret key *x* and two secret numbers *r* and *y*. For each service provider, said S_j, the registration center *RC* uses SID_j to compute a shared secret key $h(SID_j||y)$ between *RC* and S_j, and then sends $h(SID_j||y)$ to the service provider S_j via a secure channel. Hsiang and Shih's scheme mainly consists of three phases: the registration phase, the login phase, as well as the mutual authentication and key agreement phase. We briefly review these phases as follows:

[***Registration phase***]

1. U_i selects a password PW_i and a random number *b*. Then, U_i computes $h(b\oplus PW_i)$ and sends ID_i and $h(b\oplus PW_i)$ to *RC* through a secure channel.
2. *RC* computes $(T_i, V_i, A_i, B_i, R_i, H_i)$, where $T_i=h(ID_i||x)$, $V_i=T_i\oplus h(ID_i||h(b\oplus PW_i))$, $A_i=h(h(b\oplus PW_i)||r)\oplus h(x\oplus r)$, $B_i=A_i\oplus h(b\oplus PW_i)$, $R_i=h(h(b\oplus PW_i)||r)$, and $H_i=h(T_i)$. *RC* stores $<V_i, B_i, H_i, R_i, h(\cdot)>$ into a smart card and issues it to the user U_i via a secure channel.

Without loss of generality, assume that U_i wants to login the service provider S_j. The login phase as well as mutual authentication and key agreement phase are depicted in Figure 1.

[***Login phase***]

In the login phase, U_i keys his/her ID_i, PW_i and the server identity SID_j to the smart card, and then the smart card performs the following steps.

1. The smart card computes $T_i = V_i\oplus h(ID_i||h(b\oplus PW_i))$ and $H_i^*=h(T_i)$, then the smart card checks whether H_i^* is equal to H_i . If it holds, the legitimacy of the cardholder can be assured; otherwise the login request is rejected.
2. The smart card generates a nonce N_i and computes
   $A_i=B_i\oplus h(b\oplus PW_i)$,
   $CID_i = h(b\oplus PW_i)\oplus h(T_i||A_i||N_i)$,
   $P_{ij} = T_i\oplus h(A_i||N_i||SID_j)$,
   $Q_i = h(B_i||A_i||N_i)$,
   $D_i = R_i\oplus SID_j\oplus N_i$,
   and $C_0 = h(A_i||N_i+1||SID_j)$. Then the smart card sends $<CID_i, P_{ji}, Q_i, D_i, C_0, N_i>$ to the server S_j.

[***Mutual authentication and key agreement phase***]

Upon receiving the login request message $<CID_i, P_{ji}, Q_i, D_i, C_0, N_i>$, the service provider S_j authenticates the user U_i as follows.

1. S_j generates a nonce N_{jr} and computes $M_{jr} = h(SID_j\ ||\ y)\oplus N_{jr}$, and then sends the message $<M_{jr}, SID_j, D_i, C_0, N_i>$ to the registration center *RC*.
2. Upon receiving $<M_{jr}, SID_j, D_i, C_0, N_i>$, *RC* computes
   $N_{jr}' = M_{jr}\oplus h(SID_j||y)$,
   $R_i' = D_i\oplus SID_j\oplus N_i$,
   and $A_i' = R_i'\oplus h(x\oplus r)$.
   Then *RC* checks whether $h(A_i'||N_i+1||SID_j)$ is equal to C_0 or not. If it does not hold, *RC* rejects the request and terminates the session.
3. *RC* chooses $N_{rj}\in_R Z_q^*$ and computes (C_1, C_2), where $C_1 = h(N_{jr}'||h(SID_j||y)||N_{rj})$ and
   $C_2 = A_i\oplus h(h(SID_j||y)||N_{jr}')$.
   Then *RC* sends $<C_1, C_2, N_{rj}>$ to S_j.

4. Upon receiving the message $< C_1, C_2, N_{rj} >$, the server S_j checks whether $h(N_{jr}||h(SID_j||y)||N_{rj})$ is equal to C_1 or not. If it does not hold, the server S_j terminates the session.
5. The server S_j computes (A_i, T_i, $h(b \oplus PW_i)$, B_i), where $A_i = C_2 \oplus h(h(SID_j||y)||N_{rj})$,
$T_i = P_{ij} \oplus h(A_i||N_i||SID_j)$,
$h(b \oplus PW_i) = CID_i \oplus h(T_i||A_i||N_i)$,
and
$B_i = A_i \oplus h(b \oplus PW_i)$.
S_j checks whether Q_i is equal to $h(B_i||A_i||N_i)$ or not. If it does not hold, the server S_j rejects the login request and terminates the session.

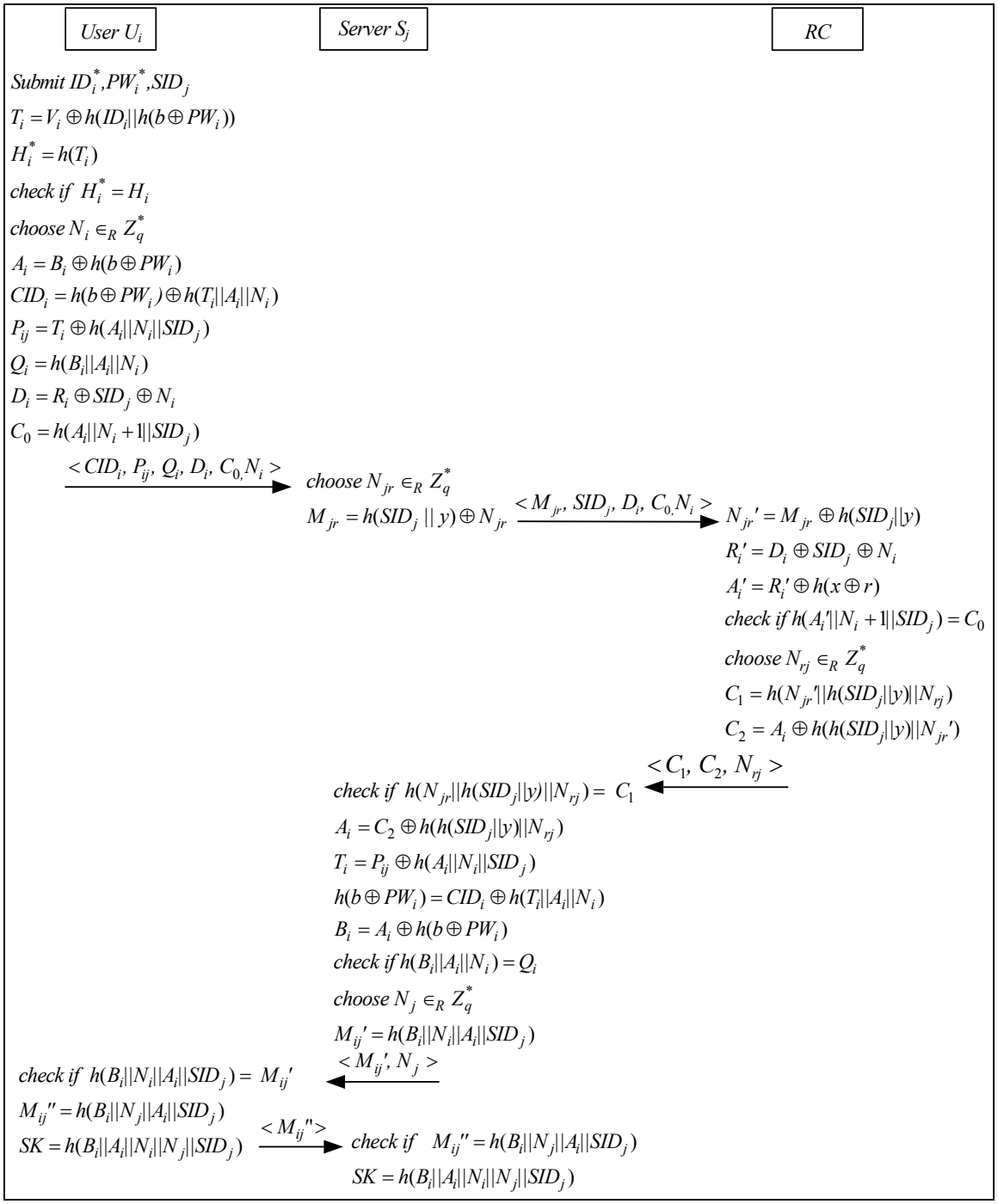

Fig.1. the login, mutual authentication and key agreement phases of Hsiang-Shih scheme

6. The server S_j chooses $N_j \in_R Z_q^*$ and computes $M_{ij} = h(B_i||N_i||A_i||SID_j)$. S_j sends $< M_{ij}', N_j >$ to the user U_i.
7. Upon receiving (M_{ij}', N_j), U_i checks whether M_{ij}' is equal to $h(B_i||N_i||A_i||SID_j)$ or not. If it does not hold, U_i interrupts the connection.
8. U_i computes $M_{ij}'' = h(B_i||N_j||A_i||SID_j)$, and then sends it to the server S_j.
9. Upon receiving the message M_{ij}'', the server S_j checks whether M_{ij}'' is equal to $h(B_i||N_j||A_i||SID_j)$ or not. If it holds, the legality of the user U_i can be assured.

After finishing the mutual authentication and key agreement phase, both the user U_i and the server S_j can compute the common session key $SK = h(B_i||A_i||N_i||N_j||SID_j)$.

B. Attacks on Hsiang and Shih's scheme

In this subsection, we demonstrate that Hsiang and Shih's scheme is vulnerable to an insider attack and a server-spoofing attack. We show that any legal user can compute a secret value $h(x \oplus r)$. Meanwhile, a server can also compute $h(x \oplus r)$ when any user has ever login the server. Then we will show that anyone who has $h(x \oplus r)$ can compute any session keys between users and servers, as well as counterfeit the other servers.

Since U_i is a legal user and has $<h(b \oplus PW_i), V_i, B_i, R_i, H_i >$, U_i can obtain $h(x \oplus r)$ by computing
$A_i = B_i \oplus h(b \oplus PW_i)$,
and
$h(x \oplus r) = A_i \oplus R_i = B_i \oplus h(b \oplus PW_i) \oplus R_i$.

At the same reason, suppose that there exists a user U_i who had ever login the server S_j, so S_j can get $<CID_i, P_{ij}, Q_i, D_i, C_0, N_i>$, $<C_1, C_2, N_{rj}>$ and M_{ij}'. Then, S_j can obtain $h(x \oplus r) = A_i \oplus R_i$ by computing $A_i = C_2 \oplus h(h(SID_j||y) \oplus N_{rj})$, $R_i = D_i \oplus SID_j \oplus N_i$.

According to the descriptions above, we have showed that any legal users or any servers can obtain $h(x \oplus r)$. In the following, we show that any attacker with $h(x \oplus r)$ can perform an insider attack and a server-spoofing attack.

(*i*) Insider attack

Here, we show that Hsiang and Shih's scheme cannot resist the insider attack. Without loss of generality, suppose that the malicious insider U_i is a legal user and has obtained $h(x \oplus r)$. The malicious insider U_i can perform the following steps to get the session key $SK = h(B_a||A_a||N_a||N_b||SID_b)$ between the any user U_a and any server S_b.

1. U_i may intercept the transmission $< CID_a, P_{ab}, D_a, N_a, N_b>$ between the user U_a and the server S_b.
2. U_i computes $(R_a, A_a, T_a, h(b \oplus PW_a), B_a)$, where
   $R_a = D_a \oplus SID_b \oplus N_a$,
   $A_a = R_a \oplus h(x \oplus r)$,
   $T_a = P_{ab} \oplus h(A_a|| N_a||SID_b)$,
   $h(b \oplus PW_a) = CID_a \oplus h(T_a||A_a||N_a)$,
   and
   $B_a = A_a \oplus h(b \oplus PW_a)$.
   Thus, the malicious insider U_i can get the session key $SK = h(B_a||A_a||N_a||N_b||SID_b)$.

(*ii*) Server-spoofing attack

In the following, we show that any attacker with the value $h(x \oplus r)$ can counterfeit any server. Hence, Hsiang and Shih's scheme cannot resist the server-spoofing attack. Since we have shown that any legitimate user U_i can obtain $h(x \oplus r)$, the legitimate user U_i can do the following steps to impersonate any server S_b to any user U_a

1. When U_a sends $< CID_a, P_{ab}, Q_a, D_a, C_0, N_a>$ to U_i, U_i randomly chooses $N_j \in Z_q^*$ and computes $(R_a, A_a, T_a, h(b \oplus PW_a), B_a, M_{ab})$, where
   $R_a = D_a \oplus SID_b \oplus N_a$,
   $A_a = R_a \oplus h(x \oplus r)$,
   $T_a = P_{ab} \oplus h(A_a|| N_a||SID_b)$,
   $h(b \oplus PW_a) = CID_a \oplus h(T_a||A_a||N_a)$,
   $B_a = A_a \oplus h(b \oplus PW_a)$,
   and
   $M_{ab} = h(B_a||N_a||A_a||SID_b)$.
   Then, U_i sends $< M_{ab}, N_j>$ to the user U_a.
2. The user U_a will check whether $M_{ab} = h(B_a||N_a||A_a||SID_b)$ holds or not. It is clear that M_{ab} is equal to $h(B_a||N_a||A_a||SID_b)$. Hence, U_a will believe that U_i is the server S_b.

III. Analysis of Geng and Zhang's Scheme

In this section, we briefly review Geng and Zhang's scheme and then demonstrate the security weakness of their scheme.

A. Review of Geng and Zhang's scheme

We briefly present the definitions and properties of bilinear pairings, which are used in Geng and Zhang's scheme. Let G_1 be an additive cyclic group with a prime order q and G_2 be a multiplicative group with the same order q. G_1 is a subgroup of points on an elliptic curve over a finite field $E(F_p)$ and P is the generator of G_1. G_2 is a subgroup of the multiplicative group over a finite field. A bilinear pairing is a map $\hat{e}: G_1 \times G_1 \rightarrow G_2$ which satisfies the following requirements:

1. Bilinear: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all $P, Q \in G_1$ and $a, b \in Z_q^*$.
2. Non-degenerate: there exist $P, Q \in G_1$ such that $\hat{e}(P, Q) \neq 1$.
3. Computability: there is an efficient algorithm to compute $\hat{e}(P, Q)$ for all $P, Q \in G_1$.

The notations used in this scheme are summarized as follows:

- $H(\ )$: a one-way hash function $\{0, 1\}^* \rightarrow G_1$.
- $f(\ )$: a one-way hash function $\{0, 1\}^* \rightarrow Z_q^*$.
- s: the secret key of *RC*.
- Pub_{RC}: the public key of *RC*, where $Pub_{RC} = sP$.
- x_j : the secret key of S_j.
- Pub_j: the public key of S_j, where $Pub_j = x_jP$.
- ID_i: the identity of U_i.
- PW_i: the password of U_i.

Without loss of generality, suppose that the multi-server system consists of one registration center (*RC*), m users and n service providers. Geng and Zhang's scheme mainly consists of two phases, the registration phase, as well as the login and session key agreement phase. We briefly review two phases as follows:

[***Registration Phase***]

In the registration phase, a user U_i submits ID_i and $h(PW_i)$ to the registration center *RC*. Then, *RC* computes (SID_i, P_i, V_i, Ver_i), where

$SID_i = H(ID_i, ID_{RC})$,

$P_i = s \cdot SID_i$,

$V_i = P_i + H(ID_i \,||\, h(PW_i))$,

and

$Ver_i = f(P_i)$.

RC computes $AID_i = \hat{e}(H(ID_{RC}), SID_i)^{f(s)}$ and stores $< SID_i, V_i, Ver_i, AID_i, H(\ ), f(\ ) >$ into a smart card and issues it to the user U_i via a secure channel.

[***Login & Session Key Agreement Phase***]

When the user U_i wants to access the resources of the server S_j, U_i inserts the smart card and keys his/her ID_i^*, PW_i^* and the session identity SID_j. The smart card computes $P_i^* = V_i - H(ID_i^* || PW_i^*)$ and checks whether $f(P_i^*)$ is equal to Ver_i or not. If it holds, the validity of the cardholder can be assured. The login and session key agreement phase is depicted in Figure 2. The smart card (U_i) and S_j perform the following steps to achieve mutual authentication and key agreement.

1. U_i randomly chooses $r_1, N_i \in_R Z_q^*$ and computes

   $C_1 = r_1 P$,

   $CID_i = SID_i + r_1 \cdot Pub_j$,

   $h = f(N_i || C_1)$,

   and

   $W = r_1^{-1}(P_i^* + hP)$.

   Then, U_i sends the login request message $< CID_i, C_1, N_i, W >$ to the service provider S_j .
2. Upon receiving the login request message $< CID_i, C_1, N_i, W >$, the service provider S_j computes $SID_i^* = CID_i - x_j \cdot C_1$ and $h = f(N_i || C_1)$. The service provider S_j checks whether $\hat{e}(W, C_1) = \hat{e}(SID_i, Pub_{RC}) \cdot \hat{e}(P,P)^h$ holds or not. If it does not hold, S_j rejects the login request and terminates the session.
3. S_j randomly chooses $r_2 \in_R Z_q^*$ and computes (C_2, sk, AID_i^*, Ver), where

   $C_2 = r_2 P$,

   $sk = f(r_2 \cdot C_1)$,

   $AID_i^* = \hat{e}(H(ID_{RC}), SID_i)^{f(s)}$,

   and

   $Ver = f(AID_i^* || C_1 || N_i || sk)$.

Then, S_j sends $< C_2, Ver >$ to U_i.

4. Upon receiving the message $< C_2, Ver >$, U_i computes $sk^* = f(r_1 \cdot C_2)$ and checks whether $f(AID_i^* || C_1 || N_i || sk^*)$ is equal to *Ver* or not.
5. U_i computes $Ver' = f(AID_i || C_2 || N_i || sk^*)$ and sends it to the server S_j.
6. Upon receiving the message Ver', the service provider S_j checks whether Ver' is equal to $f(AID_i || C_2 || N_i || sk^*)$. Meanwhile, U_i and S_j have obtained an identical session key $sk = f(r_2 \cdot C_1) = f(r_1 \cdot C_2)$.

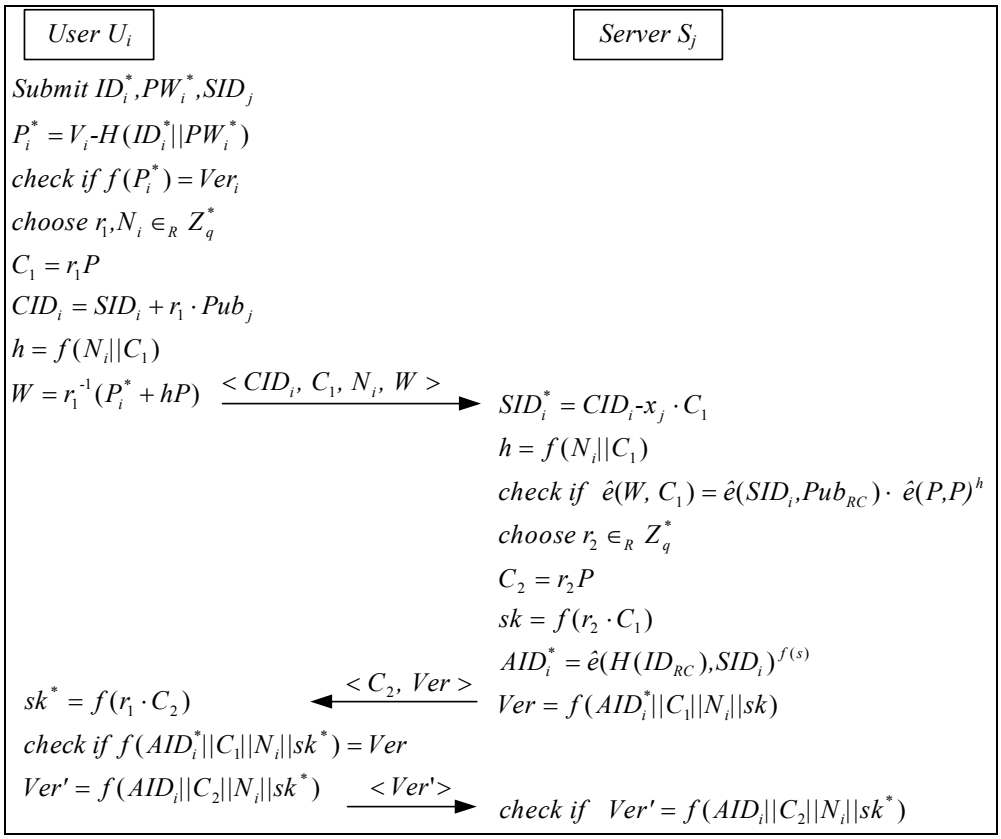

Fig.2. the login and session key agreement phase of Geng-Zhang's scheme

B. Attack on Geng and Zhang's scheme

In this subsection, we will show that Geng and Zhang's scheme is vulnerable to a user-spoofing attack, i.e., any legal user can create a new user without the registration center *RC*. The concrete scenario is presented as follows.

Let U_i be any legal user, then U_i can create a new user, said U_a, without the registration center *RC*. Since U_i has $<SID_i, V_i, Ver_i, AID_i>$ and can compute $P_i = V_i - H(ID_i || h(PW_i))$, then U_i chooses a random integer $r \in Z_q^*$ and computes $SID_a = r \cdot SID_i$, $P_a = r \cdot P_i$, $AID_a = (AID_i)^r$, $V_a = P_a + H(ID_a || h(PW_a))$ and $Ver_a = f(P_a)$ for the new spoofing user U_a.

We are going to show that the spoofing user U_a can successfully login any server, said S_j, as a legitimate user.

1. U_a randomly chooses r_1, $N_a \in Z_q^*$ and computes (C_1, CID_a, h, W), where
   $C_1 = r_1 P$,
   $CID_a = SID_a + r_1 \cdot Pub_j$,
   $h = f(N_a || C_1)$,
   and
   $W = r_1^{-1}(P_a + hP)$.
   Then, U_a sends $< CID_a, C_1, N_a, W >$ to S_j.
2. S_j computes $SID_a = CID_a - x_j \cdot C_1$, and checks if $\hat{e}(W, C_1) = \hat{e}(SID_a, Pub_{RC}) \cdot \hat{e}(P, P)^h$. It is clear

that this check will hold. Since $P_a = r \cdot P_i = r \cdot s \cdot SID_i$, we have

$$\begin{aligned}\hat{e}(W, C_1) &= \hat{e}(r_1^{-1}(P_a + hP), r_1 P) \\ &= \hat{e}(P_a + hP, P) \\ &= \hat{e}(r \cdot s \cdot SID_i + hP, P) \\ &= \hat{e}(r \cdot s \cdot SID_i, P) \cdot \hat{e}(hP, P) \\ &= \hat{e}(r \cdot SID_i, sP) \cdot \hat{e}(hP, P) \\ &= \hat{e}(SID_a, Pub_{RC}) \cdot \hat{e}(P, P)^h .\end{aligned}$$

The server S_j randomly chooses $r_2 \in Z_q^*$, and computes (C_2, sk, AID_a^*, Ver), where

$C_2 = r_2 P$,

$sk = f(r_2 \cdot C_1)$,

$AID_a^* = \hat{e}(H(ID_{RC}), SID_a)^{f(s)}$,

and

$Ver = f(AID_a^*, C_1, N_a, sk)$.

Then, S_j sends $< C_2, Ver >$ to the user U_a.

3. The user U_a computes $Ver' = f(AID_a, C_2, N_a, sk)$ and sends it to the server S_j.
4. The server S_j checks if $Ver' = f(AID_a^*, C_2, N_a, sk)$ or not. Since

$$\begin{aligned}AID_a &= (AID_i)^r \\ &= \hat{e}(H(ID_{RC}), SID_i)^{f(s) \cdot r} \\ &= \hat{e}(H(ID_{RC}), r \cdot SID_i)^{f(s)} \\ &= \hat{e}(H(ID_{RC}), SID_a)^{f(s)} \\ &= AID_a^*\end{aligned}$$

, it will pass the verification. Hence, the spoofing user U_a can successfully login any server S_j.

IV. Conclusion and Future Work

In this paper, we have shown that two dynamic ID-based remote user authentication and key agreement schemes for multi-server environment have security weaknesses. Hsiang and Shih' scheme is vulnerable to an insider attack and a server-spoofing attack. Geng and Zhang's scheme suffers from a user-spoofing attack that each legal user can create a new user without the registration center *RC*.

Recently, to develop a dynamic ID-based remote user authentication scheme for the multi-server environment has become a new research topic. However, the recently proposed schemes for this issue do not establish the attack model and provide formal security proof. Thus, they are easy to suffer from some attacks. In the future, we hope to construct the attack model and propose a provably secure dynamic ID-based remote user authentication and key agreement for the multi-server environment.

Acknowledgements

This research is partially supported by National Science Council, Taiwan, R.O.C., under contract no. NSC97-2221-E-018-010-MY3.

References

[1] C.C. Chang and J.S. Lee, "An efficient and secure multi-server password authentication scheme using smart cards", *Proceedings of the 2004 International Conference on Cyberworlds*, 2004, pp. 417-422.

[2] C.C. Chang and J.Y. Kuo, "An efficient multi-server password authenticated keys agreement scheme using smart cards with access control", *Proceedings of the 19th International Conference on Advanced Information Networking and Applications*, Vol. 2, 2005, pp. 257-260.

[3] H.Y. Chien, J.K. Jan, and Y.M. Tseng, "An efficient and practical solution to remote authentication: Smart Card," *Computers and Security*, Vol. 21, No. 4, 2002, pp. 372-375.

[4] J. Geng and L. Zhang, "A Dynamic ID-based User Authentication and Key Agreement Scheme for Multi-server Environment Using Bilinear Pairings", *Proceedings of the 2008 Workshop on Power Electronics and Intelligent Transportation System*, 2008, pp. 33-37.

[5] C. Hsiang and W.K. Shih, "Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment", *Computer Standards & Interfaces*, 2009, accepted and in press.

[6] M.S. Hwang, L.H. Li, "A new remote user authentication scheme using smart cards", *IEEE Trans. Consumer Electronics*, Vol. 46, No. 1, 2000, pp. 28– 30.

[7] W.S. Juang, "Efficient multi-server password authenticated key agreement using smart cards",

IEEE Trans. Consumer Electronics, Vol. 50, No.1, 2004, pp. 251–255.

[8] W.S. Juang, J.L. Wu, "Efficient User Authentication and Key Agreement with User Privacy Protection", *Journal of Information Science and Engineering*, Vol. 7, No. 1, 2008, pp. 120-129.

[9] M. Kim, C.K. Koc, "A Secure Hash-Based Strong-Password Authentication Protocol Using One-Time Public-Key Cryptography", *Journal of Information Science and Engineering*, Vol. 24, No. 4, 2008, pp. 1213-1227

[10] Y.C. Lee, G.K. Chang, W.C. Kuo, and J.L. Chu, "Improvement on the dynamic ID-based remote user authentication scheme", *Proceedings of Machine Learning and Cybernetics 2008*, Vol. 6, 2008, pp. 3283-3287.

[11] L.H. Li, I.C. Lin, and M.S. Hwang, "A remote password authentication scheme for multi-server architecture using neural networks", *IEEE Trans. Neural Networks*, Vol.12, No. 6, 2001, pp.1498–1504.

[12] I.E. Liao, C.C. Lee, and M.S. Hwang, "Security enhancement for a dynamic ID-based remote user authentication scheme", *Proceedings of the International Conference on Next Generation Web Services Practices*, 2005, pp.437

[13] Y.P. Liao, S.S. Wang, "A secure dynamic ID based remote user authentication scheme for multi-server environment", *Computer Standards & Interfaces*, Vol. 31, No. 1, 2009, pp.24–29

[14] J.L. Tsai, "Efficient multi-server authentication scheme based on one-way hash function without verification table", *Computers & Security*, Vol. 27, No. 3-4, 2008, pp. 115-121.

[15] Y.M. Tseng, T.Y. Wu, J.D. Wu, "A pairing-based user authentication scheme for wireless clients with smart cards", *Informatica: International Journal*, Vol. 19, No. 2, 2008, pp. 285-302.