International Journal of u- and e- Service, Science and Technology
Vol.8, No. 7 (2015), pp.127-136
http://dx.doi.org/10.14257/ijunesst.2015.8.7.13

Security Enhanced Dynamic ID based Remote User Authentication Scheme for Multi-Server Environments

Jun-Sub Kim[1] and Jin Kwak[2*]

[1]*IT Convergence Research Institute, Sungkyunkwan University, Korea*
[2]*Department of Information and Compute Engineering, Ajou University, Korea*
[1]*jskim.isaa@gmail.com*, [2]*security@ajou.ac.kr*

Abstract

Multi-server environments are that the user registers the single registration server and since the user uses the service to authenticate on multi-server. For this, many user authentication schemes have been proposed for multi-server environments. In 2013, Li, et al., proposed dynamic ID based remote user authentication scheme for multi-server environments. Unfortunately, their scheme is vulnerable to forgery attacks and replay attacks. In this paper, we analyze the security vulnerabilities of Li et al.'s scheme, and propose dynamic ID based remote user authentication scheme for multi-server environments. The proposed scheme ensures the safety to various attacks such as forgery attacks and replay attacks. Just like the existing schemes, our scheme is efficient at using the hash function and exclusive-OR operation.

Keywords: *Dynamic ID, Mutual authentication, Password, Multi-Server Environment*

1. Introduction

User authentication is used to authenticate a legitimate user through an insecure channel. According to the use environment, user authentication can be divided into user authentication in single-server environments and user authentication in multi-server environments. For user authentication in multi-server environments, users can register at a registration center to access a server that is associated with the registration center. Many user authentication schemes have been proposed for multi-server environments [1–12].

In 2009, Hsiang and Shih proposed a secure dynamic-ID-based remote user authentication scheme for multi-server environments [7]. They claimed that their scheme is secure and more efficient than Liao and Wang's scheme. Unfortunately, Lee, *et al.*, pointed out that Hsiang and Shih's scheme is vulnerable to masquerade attacks and server spoofing attacks, is not easily reparable, and cannot provide mutual authentication [8]. To solve these problems, Lee, *et al.*, proposed an improvement to Hsiang and Shih's scheme. Recently, Li, *et al.*, showed that Lee, *et al.'s* scheme does not provide authentication and is vulnerable to forgery attacks and server spoofing attacks [9]. In addition, Li *et al.*, proposed a new scheme based on Lee, *et al.'s* scheme. However, their scheme is vulnerable to forgery attacks and replay attacks. In this paper, we analyze the security vulnerabilities of Li, *et al.'s* dynamic-ID-based remote user authentication scheme and then we propose a security enhanced dynamic ID based remote user authentication scheme for multi-server environments.

This study is organized as follows: Section 2 describes a brief review of Li, *et al.'s* scheme, and Section 3 analyze the security vulnerabilities of Li et al.'s scheme. In Section 4, we propose a security enhanced dynamic ID based remote user authentication scheme for multi-server environments. In Section 5, we analyze our proposed scheme of the security requirements and performance. Finally, Section 6 presents our conclusion.

ISSN: 2005-4246 IJUNESST

2. Review of Li, *et al.'s* Scheme

Registration Phase

Step R1. $U_i \rightarrow RC : \{ID_i, A_i\}$
The user U_i selects identity ID_i and password PW_i and computes $A_i = h(b \oplus PW_i)$, where b is a random number generated by U_i. U_i sends ID_i and A_i to RC through a secure channel.

Step R2. After receiving a registration message from U_i, RC computes $h(x \| y)$ and $h(SID_j \| h(y))$ using x, y, and SID_j, where x and y are the master secret key and a secret number selected by RC, and SID_j is the identity of the server.

Step R3. $RC \rightarrow U_i : \{smart\ \ card\ [C_i, D_i, E_i, h(\cdot), h(y)]\}$
RC sends $h(x \| y)$ and $h(SID_j \| h(y))$ to server S_j through a secure channel and computes $B_i = h(ID_i \| x)$, $C_i = h(ID_i \| h(y) \| A_i)$, $D_i = h(B_i \| h(x \| y))$, and $E_i = B_i \oplus h(x \| y)$. Then RC issues the smart card containing $[C_i, D_i, E_i, h(\cdot), h(y)]$ and delivers it to U_i through a secure channel.

Step R4. U_i enters b into his/her smart card, and the smart card contains $[C_i, D_i, E_i, b, h(\cdot), h(y)]$.

Login phase

Step L1. The user U_i inserts his/her smart card into the device and inputs ID_i and PW_i. Then the smart card computes $A_i = h(b \oplus PW_i)$ and $C_i^* = h(ID_i \| h(y) \| A_i)$ and checks whether C_i^* is equal to C_i. If they are not equal, the procedure is terminated.

Step L2. Otherwise, the smart card generates a random nonce N_i and computes $P_{ij} = E_i \oplus h(h(SID_j \| h(y)) \| N_i)$, $CID_i = A_i \oplus h(D_i \| SID_j \| N_i)$, $M_1 = h(P_{ij} \| CID_i \| D_i \| N_i)$, and $M_2 = h(SID_j \| h(y)) \oplus N_i$.

Step L3. $U_i \rightarrow S_j : \{P_{ij}, CID_i, M_1, M_2\}$
U_i sends P_{ij}, CID_i, M_1, and M_2 to S_j.

Verification Phase

Step V1. After receiving the login request message from U_i, S_j computes $N_i = h(SID_j \| h(y)) \oplus M_2$, $E_i = P_{ij} \oplus h(h(SID_j \| h(y)) \| N_i)$, $B_i = E_i \oplus h(x \| y)$, $D_i = h(B_i \| h(x \| y))$, $A_i = CID_i \oplus h(D_i \| SID_j \| N_i)$, and $M_1^* = h(P_{ij} \| CID_i \| D_i \| N_i)$. Then, S_j checks whether M_1^* is equal to M_1. If they are equal, S_j accepts the login request to U_i.

Step V2. Then, S_j generates a random nonce N_j and computes $M_3 = h(D_i \| A_i \| N_j \| SID_j)$ and $M_4 = A_i \oplus N_i \oplus N_j$, and sends M_3 and M_4 to U_i.

Step V3. After receiving the authentication message from S_j, U_i computes $N_j = A_i \oplus N_i \oplus M_4$ and $M_3^* = h(D_i \| A_i \| N_j \| SID_j)$. Then, U_i checks whether M_3^* is equal to M_3. If they are equal, U_i authenticates S_j. Next, U_i computes $M_5 = h(D_i \| A_i \| N_i \| SID_j)$ and sends M_5 to S_j.

Step V4. After receiving the mutual authentication message from U_i, S_j computes $M_5^* = h(D_i \| A_i \| N_i \| SID_j)$ and checks whether M_5^* is equal to M_5. If they are equal, the procedure is terminated. Otherwise, S_j authenticates U_i, and the mutual authentication is completed.

After the mutual authentication, U_i and S_j computes $SK = h(D_i \| A_i \| N_i \| N_j \| SID_j)$ for future secure communication.

Vulnerability of Li, *et al.'s* Scheme

Forgery Attack

Assume that an attacker Z is a legal user of the system, stolen U_i's smart card, and eavesdrops the communication between U_i and S_j. Then, Z extracts the parameters $[C_i, D_i, E_i, b_i, h(\cdot), h(y)]$ stored in U_i's smart card and computes $N_i = h(SID_j | h(y)) \oplus M_2$ and $A_i = CID_i \oplus h(D_i \| SID_j \| N_i)$.

Step 1. Z inserts his/her smart card into the device and inputs ID_z and PW_z. The smart card computes $A_z = h(b_z \oplus PW_z)$ and $C_z^* = h(ID_z \| h(y) \| A_z)$ and checks whether C_z^* is equal to C_z. Z generates a random number N_z and computes $P_{ik} = E_i \oplus h(h(SID_k \| h(y)) \| N_z)$, $CID_i = A_i \oplus h(D_i \| SID_k \| N_z)$, $M_1 = h(P_{ik} \| CID_i \| D_i \| N_z)$, and $M_2 = h(SID_k \| h(y)) \oplus N_z$.

Step 2. Then, Z sends the forged login request message $\{P_{ik}, CID_i, M_1, M_2\}$ to server S_k.

Step 3. After receiving the login request message $\{P_{ik}, CID_i, M_1, M_2\}$, S_k computes $N_z = h(SID_k \| h(y)) \oplus M_2$, $E_i = P_{ik} \oplus h(h(SID_k \| h(y)) \| N_z)$, $B_i = E_i \oplus h(x \| y)$, $D_i = h(B_i \| h(x \| y))$, $A_i = CID_i \oplus h(D_i \| SID_k \| N_z)$, and $M_1^* = h(P_{ik} \| CID_i \| D_i \| N_z)$, and checks whether M_1^* is equal to M_1. If they are equal, S_k accepts the login request to Z.

Step 4. S_k generates a random nonce N_k, computes $M_3 = h(D_i \| A_i \| N_k \| SID_k)$ and $M_4 = A_i \oplus N_z \oplus N_k$, and sends M_3 and M_4 to Z.

Step 5. After receiving the authentication message $\{M_3, M_4\}$, Z computes $N_k = M_4 \oplus A_i \oplus N_z$ and $M_3^* = h(D_i \| A_i \| N_k \| SID_k)$ and authenticates S_k because M_3^* is equal to M_3. Then, Z computes $M_5 = h(D_i \| A_i \| N_z \| SID_k)$ and sends M_5 to S_k.

Step 6. After receiving the mutual authentication message $\{M_5\}$, S_k computes $M_5^* = h(D_i \| A_i \| N_z \| SID_k)$ and checks whether M_5^* is equal to M_5. If they are equal,

S_k authenticates Z.

Therefore, Li, *et al.'s* scheme is vulnerable to forgery attack.

Replay Attack

Assume that an attacker Z eavesdrops the communication message $\{P_{ij}, CID_i, M_1, M_2, M_3, M_4, M_5\}$ between U_i and S_j. Then, Z sends the previous login request message $\{P_{ij}, CID_i, M_1, M_2\}$ to S_j.

Step 1. After receiving the login request message $\{P_{ij}, CID_i, M_1, M_2,\}$, S_j computes $N_i = h(SID_j \| h(y)) \oplus M_2$, $E_i = P_{ij} \oplus h(h(SID_j \| h(y)) \| N_i)$, $B_i = E_i \oplus h(x \| y)$, $D_i = h(B_i \| h(x \| y))$, $A_i = CID_i \oplus h(D_i \| SID_j \| N_i)$, and $M_1^* = h(P_{ij} \| CID_i \| D_i \| N_i)$ and checks whether M_1^* is equal to M_1. If they are equal, S_j accepts the login request to Z.

Step 2. S_j generates a random nonce N_j^* and computes $M_3 = h(D_i \| A_i \| N_j^* \| SID_j)$ and $M_4 = A_i \oplus N_z \oplus N_j^*$.

Step 3. Then, S_j sends M_3 and M_4 to Z.

Step 4. After receiving the authentication message $\{M_3, M_4\}$, Z checks whether M_3^* is equal to M_3, where M_3^* is the same value with received M_3 from S_j. If they are equal, Z authenticates S_j and sends M_5 to S_j.

Step 5. After previously receiving the mutual authentication message $\{M_5\}$, S_j computes $M_5^* = h(D_i \| A_i \| N_i \| SID_j)$ and checks whether M_5^* is equal to M_5. If they are equal, S_j authenticates Z.

Therefore, Li, *et al.'s* scheme is vulnerable to replay attacks.

The Proposed Scheme

In this section, we propose security enhanced dynamic ID based remote user authentication scheme for multi-server environments. Registration center RC selects the master secret key x and a secret number y, computes $h(x \| y)$ and $h(SID_j \| h(y))$, and then shares them with server S_j though a secure channel. This scheme consists of four phases: registration phase, login phase, verification phase, and password change phase.

Notation

Table 1 shows the notation used describe our proposed scheme.

Table 1. Notation of or Scheme

Notation	Description
U_i	The ith user
S_j	The jth server
RC	Registration center
ID_i	Identity of U_i
PW_i	Password of U_i
SID_j	Identity of S_j
CID_i	Dynamic ID of U_i
x	Master secret key of registration center
y	Secret number of registration center
$h(\cdot)$	A one-way hash function
$\oplus$	Exclusive OR operation
$\|\|$	Concatenation operation
$A \rightarrow B : X$	X is transmitted from A to B

Registration Phase

When a new user U_i wants to register with RC, he/she performs the following steps:

Step R1. $U_i \rightarrow RC : \{ID_i, h(ID_i \| PW_i), h(b \oplus PW_i)\}$

U_i selects identity ID_i and password PW_i, and computes $h(ID_i \| PW_i)$ and $h(b \oplus PW_i)$, where b is a random number generated by U_i. Then U_i sends ID_i, $h(ID_i \| PW_i)$, and $h(b \oplus PW_i)$ to RC for registration through a secure channel.

Step R2. $RC \rightarrow U_i : \{smart\ \ card\ \ [A_i, C_i, D_i, h(\cdot), h(y)]\}$

RC computes the followings:

$A_i = h(h(ID_i \| PW_i) \| h(b \oplus PW_i))$

$B_i = h(x \| ID_i)$

$D_i = h(ID_i \| PW_i) \oplus h(B_i \| h(x \| y))$

$E_i = B_i \oplus h(x \| y)$.

Then, RC issues a smart card containing $[A_i, D_i, E_i, h(\cdot), h(y)]$ and delivers it to U_i through a secure channel.

Step R3. U_i enters b into his/her smart card and, and the smart card contains $[A_i, D_i, E_i, b, h(\cdot), h(y)]$.

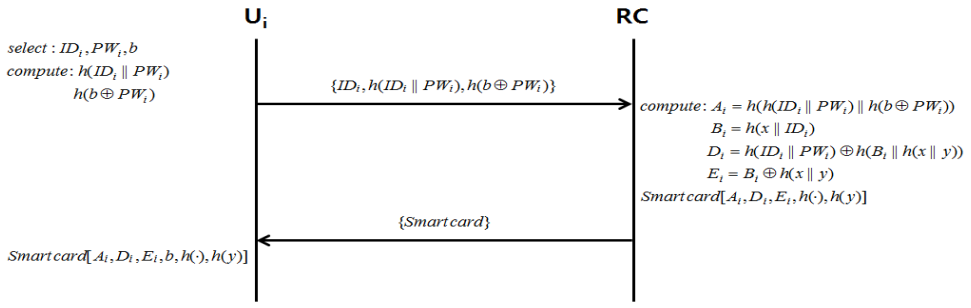

Figure 1. Registration Phase in our Scheme

Login Phase

Step L1. U_i inserts his/her smart card into the device and inputs identity ID_i and password PW_i. The smart card computes $h(ID_i \| PW_i)$, $h(b \oplus PW_i)$, and $A_i^* = h(h(ID_i \| PW_i) \| h(b \oplus PW_i))$ and checks whether A_i^* is equal to A_i. If they are not equal, the procedure is terminated. Otherwise, U_i proceeds the following steps.

Step L2. The smart card computes the followings:

$$CID_i = h(b \oplus PW_i) \oplus h(h(ID_i \| PW_i) \| SID_j \| N_i)$$

$$C_1 = h(SID_j \| h(y)) \oplus N_i$$

$$C_2 = h(h(SID_j \| h(y)) \| N_i) \oplus E_i$$

$$C_3 = h(E_i \| h(SID_j \| h(y)) \| N_i) \oplus D_i$$

$$C_4 = h(h(b \oplus PW_i) \| CID_i \| D_i \| E_i \| N_i),$$

where N_i is a random nonce generated by U_i.

Step L3. $U_i \rightarrow S_j : \{CID_i, C_1, C_2, C_3, C_4\}$

U_i sends CID_i, C_1, C_2, C_3, and C_4 to S_j.

Verification Phase

Step V1. S_j computes the followings:

$$N_i = C_1 \oplus h(SID_j \| h(y))$$

$$E_i = C_2 \oplus h(h(SID_j \| h(y)) \| N_i)$$

$$B_i = E_i \oplus h(x \| y)$$

$$D_i = C_3 \oplus h(E_i \| h(SID_j \| h(y)) \| N_i)$$

$$h(ID_i \| PW_i) = D_i \oplus h(B_i \| h(x \| y))$$

$$h(b \oplus PW_i) = CID_i \oplus h(h(ID_i \| PW_i) \| SID_j \| N_i)$$

$$C_4' = h(h(b \oplus PW_i) \| CID_i \| D_i \| E_i \| N_i).$$

Step V2. S_j checks whether C_4' is equal to C_4. If they are equal, S_j generates random nonce N_j. Then, S_j computes the followings:

$$C_5 = h(b \oplus PW_i) \oplus N_i \oplus N_j$$

$$C_6 = h(h(b \oplus PW_i) \| h(ID_i \| PW_i) \| SID_j \| N_i),$$

and sends C_5 and C_6 to U_i.

Step V3. U_i computes the followings:

$$N_j = C_5 \oplus h(b \oplus PW_i) \oplus N_i$$

$$C_6' = h(h(b \oplus PW_i) \| h(ID_i \| PW_i) \| SID_j \| N_i),$$

and checks whether C_6' is equal to C_6. If they are equal, U_i authenticates S_j. Then, U_i computes $C_7 = h(h(b \oplus PW_i) \| h(ID_i \| PW_i) \| SID_j \| N_j)$ and sends C_7 to S_j.

Step V4. S_j computes $C_7' = h(h(b \oplus PW_i) \| h(ID_i \| PW_i) \| SID_j \| N_j)$ and checks whether C_7' is equal to C_7. If they are not equal, the procedure is terminated. Otherwise, S_j authenticates U_i and the mutual authentication is completed.

After the mutual authentication, U_i and S_j computes $SK = h(h(b \oplus PW_i) \| h(ID_i \| PW_i) \| SID_j \| N_i \| N_j)$ for future secure communication.

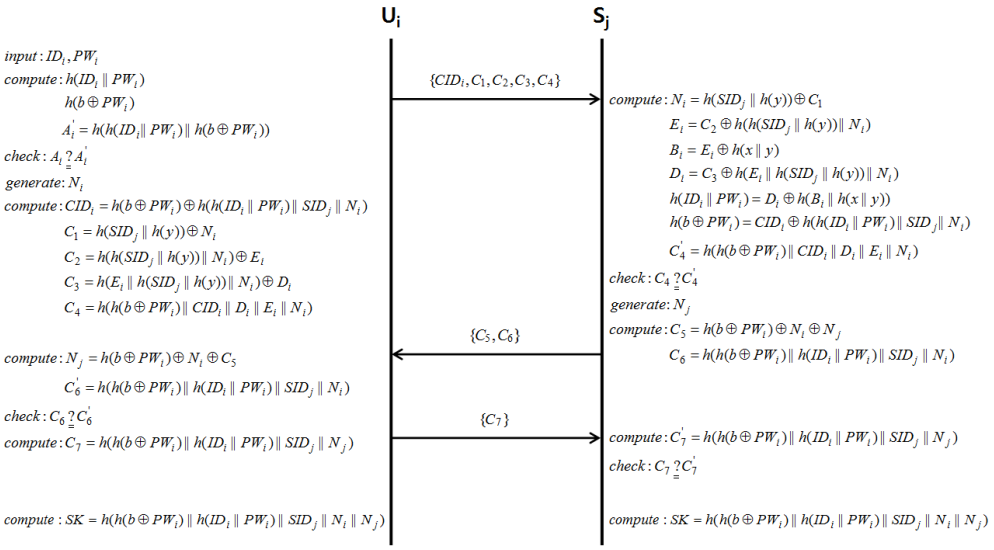

Figure 2. Login and Verification Phase in our Scheme

Password Change Phase

Step P1. U_i inserts his/her smart card into the device and inputs identity ID_i and password PW_i.

Step P2. The smart card computes $h(ID_i \| PW_i)$, $h(b \oplus PW_i)$, and $A_i^* = h(h(ID_i \| PW_i) \| h(b \oplus PW_i))$, and checks whether A_i^* is equal to A_i. If they are not equal, the procedure is terminated. Otherwise, U_i inputs a new password PW_i^{new} and a new random number b^{new}.

Step P3. The smart card computes $h(ID_i \| PW_i^{new})$, $h(b^{new} \| PW_i^{new})$, $A_i^{new} = h(h(ID_i \| PW_i^{new}) \| h(b^{new} \| PW_i^{new}))$, and $D_i^{new} = D_i \oplus h(ID_i \| PW_i) \oplus h(ID_i \| PW_i^{new})$.

Step P4. Finally, the smart card replaces A_i with A_i^{new}, and D_i with D_i^{new}.

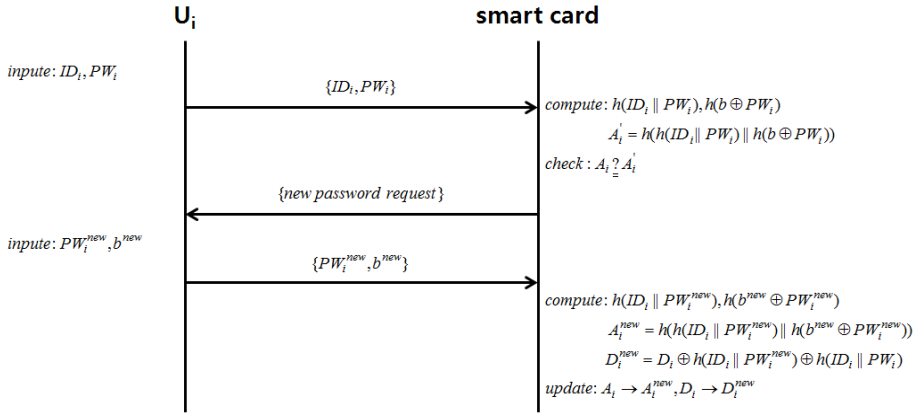

Figure 3. Password Change Phase in our Scheme

3. Analyses

Security Analysis

Table 2 compares the security of existing schemes with our proposed scheme. Our scheme has the following security properties:

Known-key Secrecy: If an attacker obtains the session key SK in the previous session, he/she cannot compute session key in later session. As the nature of a one-way hash function, an attacker cannot obtain $h(ID_i \| PW_i)$ and $h(b \oplus PW_i)$ in session key. Also if an attacker are stolen U_i's smart card or eavesdrops the previous message, he/she compute the session key from it.

Forward Secrecy: If an attacker obtains the master secret key x, he/she cannot compute the session key of the previous session. The reason cannot know b, ID_i, and PW_i. Although an attacker are stolen U_i's smart card or eavesdrops the previous message, he/she compute the previous session key without knowing b, ID_i, and PW_i.

Replay Attack: Assume that an attacker eavesdrops on the login request message $\{CID_i, C_1, C_2, C_3, C_4, C_5, C_6, C_7\}$ for the previous session. An attacker is replaying the eavesdropping message to authenticate, but he/she cannot authenticate from the server.

Forgery Attack: Because an attacker cannot know b, ID_i, PW_i, N_i, $h(y)$, E_i, and D_i, he/she cannot compute the login request message $\{CID_i, C_1, C_2, C_3, C_4\}$. Although an attacker are a legitimate user of the system, he/she cannot compute the login request message because of without knowing b, PW_i, and ID_i. Even though an attacker also obtain user's smart card, he/she cannot compute the login request message because of without knowing ID_i and PW_i.

Server Spoofing Attack and Registration Center Spoofing Attack: When an attacker masquerade as a legitimate server, he/she cannot obtain $h(ID_i \| PW_i)$ and $h(b \oplus PW_i)$ because of without knowing $h(x \| y)$. Also because a legitimate server cannot know $h(y)$, he/she cannot masquerade as the other server. An attacker cannot masquerade as a legitimate registration center because of without knowing x and y.

Stolen Smart Card Attack: Assume that an attacker obtains U_i's smart card or extracts information of U_i's smart card. But an attacker cannot compute the login request message because of without knowing ID_i and PW_i. Also an attacker cannot obtain ID_i and PW_i through the extracted information. The reason cannot obtain ID_i and PW_i from the extracted information by the nature of a one-way hash function.

Mutual Authentication: The user can authenticate the server by checking C_6 and the server can authenticate the user by checking C_7.

Table 2. Security Analysis of the Compared Schemes

	Proposed scheme	Hsiang, *et al.* [7]	Lee, *et al.* [8]	Li, *et al.* [9]
Known-key secrecy	Yes	Yes	Yes	Yes
Forward secrecy	Yes	Yes	Yes	Yes
Replay attack	Yes	Yes	Yes	No
Forgery attack	Yes	No	No	No
Server spoofing attack and registration center spoofing attack	Yes	No	No	Yes
Stolen smart card attack	Yes	Yes	Yes	Yes
Mutual authentication	Yes	No	No	Yes

4. Performance Analysis

Table 3 compares the performance of existing schemes with our proposed scheme. Figure 4 show the measured results for users performing the login and verification phase. As shown in Figure 4, our proposed scheme incurs less computational time than Hsiang, *et al.'s* scheme and incurs little more computational time as Lee, *et al.'s* scheme and Li, *et al.'s* scheme. However our scheme is more secure against various attacks.

Table 3. Performance Analysis of the Compared Schemes

Scheme		Proposed scheme	Hsiang, *et al.* [7]	Lee, *et al.* [8]	Li, *et al.* [9]
Registration Phase	User	2T(h)+1T(⊕)	1T(h)+1T(⊕)	1T(h)+1T(⊕)	1T(h)+1T(⊕)
	RC	6T(T)+2T(⊕)	5T(T)+4T(⊕)	6T(T)+1T(⊕)	6T(T)+1T(⊕)
Login & Verification Phase	User	11T(h)+7T(⊕)	10T(h)+7T(⊕)	10T(h)+4T(⊕)	9T(h)+6T(⊕)
	Server	8T(h)+8T(⊕)	9T(h)+6T(⊕)	8T(h)+2T(⊕)	7T(h)+6T(⊕)
	RC	·	3T(h)+6T(⊕)	·	·
Total		27T(h)+18T(⊕)	28T(h)+24T(⊕)	25T(h)+8T(⊕)	23T(h)+14T(⊕)

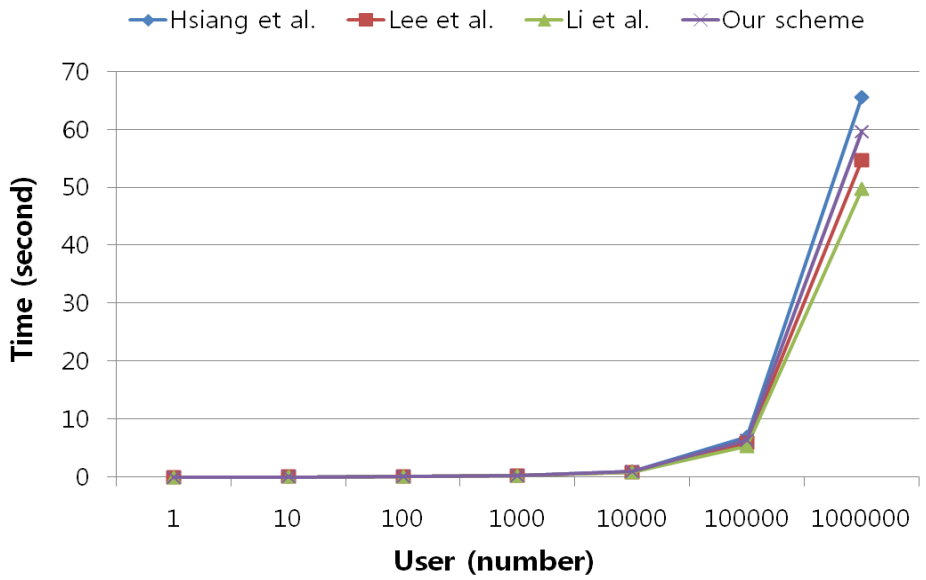

Figure 4. Analysis of Time According To Number of Users

5. Conclusions

In this paper, we examined the security vulnerabilities of Li, *et al.'s* scheme and proposed a security enhanced dynamic ID based remote user authentication scheme for multi-server environments. Our scheme provides mutual authentication and key establishment between user and server. In addition, our scheme has been proved that security to various attacks. Although our scheme incurs a little more computational costs than Lee, *et al.'s* scheme and Li, *et al.'s* scheme, it is more secure against various attacks than Lee, *et al.'s* scheme and Li, *et al.'s* scheme.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2014R1A2A1A11050818).

References

[1] W. Tsaur, "A Flexible User Authentication Scheme for Multi-server Internet Services", Lecture Notes in Computer Science, vol. 2093, **(2001)**, pp. 174-183.
[2] L. Li, I. Lin and M. Hwang, "A remote password authentication scheme for multiserver architecture using neural networks", IEEE Transactions on Neural Networks, vol. 12, no. 6, **(2001)**, pp. 1498-1504.
[3] I. Lin, M. Hwang and L. Li, "A new remote user authentication scheme for multi-server architecture", Future Generation Computer Systems, vol. 19, no. 1, **(2003)**, pp. 13-22.
[4] W. Juang, "Efficient multi-server password authenticated key agreement using smart cards", IEEE Transactions on Consumer Electronics, vol. 50, no. 1, **(2004)**, pp. 251-255.
[5] C. Chang and J. Lee, "An efficient and secure multi-server password authentication scheme using smart cards", Proc. International Conference on Cyberworlds, **(2004)**, pp. 417-422.
[6] Y. Liao and 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.
[7] H. Hsiang and W. Shih, "Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment", Computer Standards & Interfaces, vol. 31, no. 6, **(2009)**, pp. 1118-1123.
[8] C. Lee, T. Lin and R. Chang, "A secure dynamic ID based remote user authentication scheme for multi-server environment using smart cards", Expert Systems with Applications, vol. 38, no. 11, **(2011)**, pp. 13863-13870.
[9] X. Li, J. Ma, W. Wang, Y. Xiong and J. Zhang, "A novel smart card and dynamic ID based remote user authentication scheme for multi-server environments", Mathematical and Computer Modelling, vol. 58, no. 1-2, **(2013)**, pp. 85-95.
[10] C. Lee, Y. Lai and C. Li, "An Improved Secure Dynamic ID Based Remote User Authentication Scheme for Multi-Server Environment", IJSIA, vol. 6, no. 2, **(2012)**, pp. 203-209.
[11] C. Li, C. Weng and C. Fan, "Two-Factor User Authentication in Multi-Server Networks", IJSIA, vol. 6, no. 2, **(2012)**, pp. 261-267.
[12] C. Li, C. Lee, H. Mei and C. Yang, "A Password and Smart Card Based User Authentication Mechanism for Multi-Server Environments", IJFGCN, vol. 5, no. 4, **(2012)**, pp. 153-163.

