An Improved Dynamic ID-based Remote User Authentication with Key Agreement Scheme for Multi-server Environment

Zhen-Zhen Wang and Jin-Kou Ding
School of Science,
Beijing University of Posts and Telecommunications,
Beijing, China
zzwang_bupt@163.com

Zhen-Zhen Wang, Jin-Kou Ding, Zheng-Ping Jin, Hua Zhang
State Key Laboratory of
Networking and Switching Technology,
Beijing University of Posts and Telecommunications,
Beijing, China

***Abstract*—In 2011, Lee et al. improved Hsiang et al.'s scheme and proposed a security dynamic ID-based remote user authentication scheme for multi-server environment using smart cards. They claimed that their protocol is efficient and can resist several kinds of known attacks. However, we observe that Lee et al.'s scheme is still vulnerable to stolen smart card attack, malicious server attack. To remedy these security weaknesses, we propose an improved dynamic ID-based remote user authentication scheme which is immune to those attacks mentioned above. Besides, performance analysis shows that compared with other remote user authentication schemes, the proposed scheme possesses lower power consumption and lower computation cost. As a result, the proposed scheme seems to be more security and efficient, and more practical for users with portable mobile devices in multi-server environments.**

Keywords-Dynamic ID; authentication; key agreement; multi-server; smart card

I. INTRODUCTION

Remote user authentication is used for the remote server to verify the legitimacy of a remote login user over an insecure communication channel. With the rapid development of the Internet application and e-commerce technology, users are able to access the services with the portable mobile devices (smart card, PDA, note-book computer and so on) from any place at any time through the Internet. However, these conveniences will bring some security problems inevitably. Therefore, security such as user authentication is major worry in public network.

Password based authentication [1-2] is one of the simplest and widely adopted mechanisms in network environments to authenticate a legitimate user. In 1981, Lamport [3] introduced first well-known remote password authentication scheme with password table. In the scheme, a password verification table must be stored in the server. It is susceptible to the risk of modifying the verification table and vulnerable to some attacks. Besides, this would also introduce the risk and cost of protecting and managing the verification table. Thus, in order to eliminate the security problem and improve cost and efficiency, in 2000, Hwang and Li [4] proposed a password authentication protocol without password table. Following the work of Hwang and Li, many remote user authentication schemes without password table have been proposed. But these conventional password authentication schemes were designed for the single-server environment.

In fact, with the fast development of the Internet application, many network environments have been becoming multi-server based. However, the multi-server environment requires that a user only registers once at the registration center and then he/she can use all the permitted services in remote servers [5]. Therefore, many dynamic ID-based remote user authentication schemes [6-10] have been proposed for multi-server environment.

In 2009, Liao and Wang [6] proposed a dynamic identity based remote user authentication scheme for multi-server architecture. They claimed that their protocol can resist various attacks and can achieve mutual authentication. However, Hsiang and Shih [7] pointed out that scheme [6] is vulnerable to insider attack, masquerade attack, stolen smart card attack, server spoofing attack, registration center spoofing attack and also fails to provide mutual authentication. To remedy these weaknesses, Hsiang and Shih proposed an improvement over scheme [6]. Nevertheless, in 2011, Lee et al. [8] pointed out that scheme [7] is susceptible to masquerade attack, server spoofing attack, moreover, it is not easily reparable and also cannot provide mutual authentication. For this, Lee et al. proposed an improved scheme based on scheme [7] and claimed that their scheme is more secure and efficient.

However, we find that Lee et al.'s scheme [8] still is vulnerable to stolen smart card attack, and malicious server attack. Therefore, in this paper, we propose an improved dynamic ID based remote user authentication with key agreement scheme for multi-server environment to solve the aforementioned security weaknesses. In our scheme, each server stores a unique secret number, and we employ the registration center (*RC*) to help the remote server authenticate login user. Our scheme can resist the possible attacks resulting from the multi-server environment. As a result, our proposed scheme can meet the requirements which is presented in [8] for the remote user authentication scheme for multi-server environment.

The remainder of the paper is organized as follows. In section Ⅱ, we present the cryptanalysis of Lee et al.'s scheme. The proposed improved scheme is provided in section Ⅲ. In section Ⅳ, we show the corresponding

Published by Atlantis Press, Paris, France.

analysis about security and performance. Finally, conclusion is given in section Ⅴ.

TABLE I. THE NOTATIONS USED IN THIS PAPER

Notations	Descriptions
U_i	The i_{th} user
S_j	The j_{th} server
RC	The registration center
SID_j	The identity of S_j
CID_i	The dynamic ID of U_i
$h(\cdot)$	A one way hash function
$\oplus$	The bitwise XOR operation
$\|\|$	String concatenation operation
$\Rightarrow$	A secure channel
$\rightarrow$	A common channel

II. CRYPTANALYSIS OF LEE ET AL.'S SCHEME

In this section, we analyze the security flaws of scheme [8]. The details of Lee et al.'s scheme can refer to [8]. We find that their scheme cannot resist stolen smart card attack, malicious server attack. The details are shown as follows.

The notations used through this paper are summarized in Table Ⅰ.

A. Stolen smart card attack

Suppose that an adversary Z steals the user U_i's smart card, he/she can extract the information $\{V_i, B_i, H_i, b, h(\cdot), h(y)\}$ stored in the smart card. In addition, if a previously login message $\{CID_i, P_{ij}, Q_i, N_i\}$ launched by the user U_i in a public networks was also intercepted by the adversary Z, the adversary Z can compute $T_i = P_{ij} \oplus h(h(y) \| N_i \| SID_j)$, $A_i = h(T_i \| h(y) \| N_i)$ and $h(b \oplus PW_i) = CID_i \oplus h(T_i \| A_i \| N_i)$. Then the Z can impersonate the user U_i to login to the server S_j. The Z generates a random number N_z, and computes $A_i^* = h(T_i \| h(y) \| N_z)$, $CID_i^* = h(b \oplus PW_i) \oplus h(T_i \| A_i^* \| N_z)$, $P_{ij}^* = T_i \oplus h(h(y) \| N_z \| SID_j)$ and $Q_i^* = h(B_i \| A_i^* \| N_z)$. Then, Z sends the forged login request message $\{CID_i^*, P_{ij}^*, Q_i^*, N_z\}$ to the server S_j. It is obvious that the forged login request message can easily pass through the verification of S_j in step V2. Besides, in step V3, upon receiving the message $\{M_{ij}, N_j\}$ from S_j, Z can compute $M_{ij}^{'*} = h(h(T_i \| E_i \| N_i) \| N_j \| SID_j)$, then, responses with the message $\{M_{ij}^{'*}\}$ to the S_j. Obviously, the Z can be verified by S_j successfully in step V4. Therefore, the adversary Z can fool the server S_j into believing that he/she is a legitimate user. The scheme [8] cannot resist the stolen smart card attack.

B. Malicious server attack

In [8], all the servers share exactly two same secret value $h(x \| y)$ and $h(y)$. When user U_i logins to the server, each server has the ability to operate request message $\{CID_i, P_{ij}, Q_i, N_i\}$ into follow formula:

$T_i = P_{ij} \oplus h(h(y) \| N_i \| SID_j)$, $A_i = h(T_i \| h(y) \| N_i)$

$h(b \oplus PW_i) = CID_i \oplus h(T_i \| A_i \| N_i)$

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

Therefore, with the secret information $\{T_i, A_i, h(b \oplus PW_i), B_i\}$ about the user U_i, the malicious server S_j can generate a nonce $N_i^{'}$ and masquerade the user U_i to send a forged login message $\{CID_i^{'}, P_{ij}^{'}, Q_i^{'}, N_i^{'}\}$ to a legal server, say S_{j+1}, and then the server S_{j+1} can verify S_j successfully. Finally the malicious server S_j can login to S_{j+1} to access the service provided by the server S_{j+1}. Thus, scheme [8] is vulnerable to the malicious server attack.

III. OUR PROPOSED SCHEME

In this section, we propose an improved dynamic ID based remote user authentication with key agreement scheme for multi-server environment which not only remedies the security weaknesses existing in scheme [8] but also achieves more secure and efficient features. Our scheme involves three entities: the user (U_i), the server (S_j), and the registration center (RC). We assume that RC is to be trustworthy. RC chooses the master secret key x and secret number y, and then computers $h(SID_j \| y)$ and shares it with S_j in secure channel. Our scheme has four phases: registration phase, login phase, authentication and session key agreement phase, and password change phase.

Registration Phase:

Step R1: $U_i \Rightarrow RC: ID_i, E_i = h(b \oplus PW_i)$. U_i freely chooses his/her identity ID_i and password PW_i, and chooses a random number b. Then computes $E_i = h(b \oplus PW_i)$, and sends ID_i and E_i to the registration center RC for registration via a secure channel.

Step R2: RC computes $T_i = h(ID_i \| x)$, $V_i = T_i \oplus h(ID_i \| E_i)$, $B_i = T_i \oplus h(h(y \| x) \| E_i)$, $H_i = h(T_i)$.

Step R3: RC stores $\{V_i, B_i, H_i, h(\cdot), h(y)\}$ into a smart card, and issues it to the user U_i via a secure channel.

Step R4: User U_i enters b into the smart card. Finally, the smart card contains $\{V_i, B_i, H_i, b, h(\cdot), h(y)\}$.

Login Phase:

Step L1: U_i inserts his/her smart card into the smart card reader, and inputs his/her ID_i and PW_i. Then the smart card computes $E_i = h(b \oplus PW_i)$, $T_i = V_i \oplus h(ID_i \| E_i)$ and

$H_i^* = h(T_i)$, and checks whether $H_i^* = H_i$. If they are equal, smart card proceeds to the next step.

Step L2: Smart card generates a nonce N_i and computes $CID_i = E_i \oplus h(B_i \| h(y) \| N_i)$, $P_{ij} = B_i \oplus h(h(y) \| N_i \| SID_j)$ and $C_0 = h(SID_j \| T_i \| N_i \| B_i)$, then the smart card sends the login request message $\{CID_i, P_{ij}, C_0, N_i\}$ to the server S_j over a public channel.

Authentication and Session Key Agreement Phase:

Step A1: After receiving the login request message $\{CID_i, P_{ij}, C_0, N_i\}$, the server S_j generates a nonce N_{jr}, and computes $K_i = h(SID_j \| y) \oplus N_{jr}$. Then, S_j sends the login request message $\{CID_i, P_{ij}, C_0, N_i, K_i, SID_j\}$ to RC.

Step A2: On receiving the login request message from the server S_j, RC computes $N_{jr} = K_i \oplus h(SID_j \| y)$, $B_i = P_{ij} \oplus h(h(y) \| N_i \| SID_j)$ $E_i = CID_i \oplus h(B_i \| h(y) \| N_i)$, $T_i = B_i \oplus h(h(y \| x) \| E_i)$ and $C_0 = h(SID_j \| T_i \| N_i \| B_i)$. Next, RC compares C_0 with the received C_0, checks whether they are equal. If they are equal, RC authenticates the legitimacy of user U_i successfully, and further generates nonce N_{rj} and computes $C_1 = h(N_{rj} \| h(SID_j \| y) \| N_{jr})$, $C_2 = h(h(SID_j \| y) \| N_{jr}) \oplus h(T_i \| E_i \| N_i)$. Finally, RC responses the message $\{C_1, C_2, N_{rj}\}$ to S_j.

Step A3: When receiving message $\{C_1, C_2, N_{rj}\}$ from RC, the S_j computes $h(N_{rj} \| h(SID_j \| y) \| N_{jr})$ and checks whether it equals to the received C_1. If they are equal, S_j verifies the RC successfully, then generates a nonce N_j, and computes $h(T_i \| E_i \| N_i) = C_2 \oplus h(h(SID_j \| y) \| N_{jr})$, $M_{ij} = h(h(T_i \| E_i \| N_i) \| N_j \| SID_j)$. Finally, S_j sends $\{M_{ij}, N_j\}$ to U_i.

Step A4: Upon receiving the message $\{M_{ij}, N_j\}$ from S_j, U_i computes $h(h(T_i \| E_i \| N_i) \| N_j \| SID_j)$, and checks it with the received M_{ij}, if they are not equal, U_i terminates the session. Otherwise, U_i verifies the S_j and computes $M_{ij}^{'} = h(h(T_i \| E_i \| N_i) \| N_i \| SID_j)$, and then responses the message $\{M_{ij}^{'}\}$ to S_j.

Step A5: Upon receiving $\{M_{ij}^{'}\}$, the S_j computes $h(h(T_i \| E_i \| N_i) \| N_i \| SID_j)$, and checks it with the received $M_{ij}^{'}$, if they are equal, the validity of U_i is authenticated by S_j successfully.

Finally, S_j and U_i agree on a shared session key $SK = h(h(T_i \| E_i \| N_i) \| N_i \| N_j \| SID_j)$ for the future securing communications.

Password Change Phase:

Step P1: U_i inserts his/her smart card into the smart card reader, and then inputs his/her ID_i and PW_i.

Step P2: The smart card computes $E_i = h(b \oplus PW_i)$, $T_i = V_i \oplus h(ID_i \| E_i)$ and $H_i^* = h(T_i)$, and checks whether $H_i^* = H_i$. If they are equal, U_i chooses a new password PW_{new} and a new b_{new}, computes $E_{new} = h(b_{new} \oplus PW_{new})$ and $V_{new} = T_i \oplus h(ID_i \| E_{new})$, then sends ID_i and PW_{new} to RC in a secure channel.

Step P3: RC computes $B_{new} = T_i \oplus h(h(y \| x) \| E_{new})$ and sends back B_{new} to U_i.

Step P4: The smart card replaces V_i, B_i with V_{new}, B_{new}.

IV. ANALYSIS OF OUR SCHEME

A. Security analysis

In this subsection, we analyze the security features of our proposed scheme on various known cryptographic attacks and then Table II shows a comparison of the proposed scheme and other relevant schemes [7-9] about security. It demonstrates that our scheme is more secure than other related schemes.

1) Malicious user attack

In our design, the malicious privileged user U_z knows the secret value $h(y)$ shared by each user in their smart card. In addition, he/she can also intercept login message $\{CID_i, P_{ij}, C_0, N_i\}$ of other users, say U_i. Then, he/she can compute $B_i = P_{ij} \oplus h(h(y) \| N_i \| SID_j)$, $E_i = CID_i \oplus h(B_i \| h(y) \| N_i)$. Unfortunately, the secret information T_i of the user U_i is protected by the hash function, no entity can compute T_i without the knowledge of ID_i and x or $h(y \| x)$. Therefore, the attacker cannot compute correct authentication message $C_0 = h(SID_j \| T_i \| N_i \| B_i)$, this can avoid that a malicious privileged user impersonates a legal user to cheat others. Similarly, the privileged user U_z can also not impersonate the S_j or RC to fool U_i without knowing the secret key $h(T_i \| E_i \| N_i)$ and $h(y \| x)$. Therefore, our scheme can resist the malicious user attack.

2) Malicious server attack

In our protocol, the trusted third party RC is employed to help the server authenticate the remote user. As a result, the server cannot compute any secret information about user without knowing $h(y)$ and $h(y \| x)$. Therefore, the malicious privileged server cannot masquerade any user to generate valid authentication message $C_0 = h(SID_j \| T_i \| N_i \| B_i)$. This can avoid the malicious server attack.

3) Stolen smart card attack

In Lee et al.'s scheme, all the information related to the user is either stored in his/her smart card or transmitted in the open channel. However, in our scheme, the main

information T_i , which can only be computed by RC , is neither stored in the smart card nor transmitted in the open

TABLE II. FUNCTIONALITY COMPARISONS

Functionalities	Ours	[7]	[8]	[9]
Mutual authentication	Yes	No	No	Yes
Malicious user attack	Yes	Yes	No	No
Malicious server attack	Yes	No	No	Yes
Stolen smart card attack	Yes	No	No	No
Server spoofing attack	Yes	No	No	Yes

Yes: prevents the attack or provides the security property;
No: does not prevent or not provides the property.

TABLE III. PERFORMANCE COMPARISONS

Protocols	**Login Phase**	**Authentication Phase**	**Total**
Ours	$6T_h$	$14T_h$	$20T_h$
[7]	$7T_h$	$12T_h$	$19T_h$
[8]	$7T_h$	$9T_h$	$16T_h$
[9]	$6T_h$	$17T_h$	$23T_h$

channel. Even if the adversary has breached the user U_i 's information $\{V_i, B_i, H_i, b, h(\cdot), h(y)\}$ stored in the smart card, he/she cannot use parameters $\{V_i, B_i, H_i, b, h(\cdot), h(y)\}$ and the intercepted message $\{CID_i, P_{ij}, C_0, N_i\}$ in the open channel to compute the correct authentication message $C_0 = h(SID_j \| T_i \| N_i \| B_i)$, since he/she has no way to get the valid T_i . Therefore, the attacker cannot send a forged login request to cheat the server. Our protocol is secure against the stolen smart card attack.

4) Server spoofing attack

In our scheme, each server stores a unique secret number $h(SID_j \| y)$. The attacker (consist of a malicious server) cannot get the secret number $h(SID_j \| y)$ of the server S_j without knowing the secret value y , so he/she cannot compute the $h(T_i \| E_i \| N_i)$ in step A3, then he/she can also not compute the correct response authentication message $M_{ij} = h(h(T_i \| E_i \| N_i) \| N_i \| SID_j)$. Therefore, our scheme is secure against the server spoofing attack.

B. Performance analysis

This subsection compares the performance of our proposed scheme with the related schemes [7-9] and evaluates our scheme. In our scheme, we only use the one-way hash function and exclusive-OR operations to finish the entire login and authentication phases. To analyze the computational complexity of the protocols, we define the notation T_h is the time for a one-hash function. Because exclusive-OR operation requires very few computations, it is usually negligible considering its computation cost [9]. Besides, the computation cost of user registration is a one-time task, therefore, we only talk about the cost in login and authentication phase.

Table Ⅲ shows the comparison of our improved scheme and the related remote user authentication schemes [7-9]. Our protocol achieves proper balance between the security and performance. By comparing, we can conclude that our dynamic ID based remote user authentication scheme is more suitable for using in the future.

V. CONCLUSION

In this paper, we proposed an improved dynamic ID based remote user authentication with key agreement scheme for multi-server environment upon finding some certain weaknesses in Lee et al.'s scheme. We find Lee et al.'s scheme is vulnerable to stolen smart card attack, malicious server attack. However, our improved scheme can withstand various known cryptosystem attacks and we adopt the random nonce instead of the timestamps to prevent the replay attack and avoid the difficulty and cost of implementing clock synchronization. Then by the security and performance comparison, we demonstrate that our proposed scheme is more suitable in the real-life.

ACKNOWLEDGMENT

This work is supported by NSFC (Grant Nos. 61202434, 61170270, 61121061), the Fundamental Research Funds for the Central Universities (Grant Nos. 2011RC0505, 2011RCZJ15, 2012RC0612, 2011YB01).

REFERENCES

[1] G. B. Purdy, "A High Security Log-in Procedure," Communication of ACM, vol. 17, pp. 442-445, 1974.

[2] B. Menkus, "Understanding the Use of Passwords," Computers & Security, vol. 7, no. 2, pp. 132-136, 1988.

[3] L. Lamport, "Password Authentication with Insecure Communication," Communications of ACM, vol. 24, pp. 770-772, 1981.

[4] M. S. Hwang and L. H. Li, "A New Remote User Authentication Scheme Using Smart Cards," IEEE Transactions on Consumer Electronics, vol. 46, pp. 28-30, 2000.

[5] W. S. Juang, "Efficient Multi-server Password Authenticated Key Agreement Using Smart Cards," IEEE Transaction on Consumer Electronics, vol. 50, no. 1, pp. 251-255, 2004.

[6] Y. P. Liao and S. S. Wang, "A Secure Password Dynamic ID Based Remote User Authentication Scheme for Multi-server Environment," Computer Standards & Interfaces, vol. 31, pp. 24-29, 2009.

[7] H. C. Hsiang and W. K. Shih, "Improvement of The Secure Dynamic ID Based Remote User Authentication Scheme for Multi-server Environment," Computer Standard & Interfaces, vol. 31, pp. 1118-1123, 2009.

[8] C. C. Lee, T. H. Lin and R. X. Chang, "A Secure Dynamic ID Based Remote User Authentication Scheme for Multi-server Environment Using Smart Cards," Expert Systems with Applications, vol. 38, pp. 13863-13870, 2011.

[9] X. Li, Y. X. Xiong, J. Ma and W. D. Wang, "An Efficient and Security Dynamic Identity Based Authentication Protocol for Multi-server Architecture Using Smart Cards," Journal of Network and Computer Applications, vol. 35, pp. 763-769, 2012.

[10] Y. P. Liao and C. M. Hsiao, "A Novel Multi-server Remote User Authentication Scheme Using Self-certified Public Keys for Mobile Clients," Future Generation Computer Systems, article in press, 2012.