Comments on a secure dynamic ID-based remote user authentication scheme for multi-server environment using smart cards

Debiao He

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

Email: hedebiao@163.com

Tel: +008615307184927

Abstract: The security of a dynamic ID-based remote user authentication scheme for multi-server environment using smart cards proposed by Lee et al. [Lee, C-C., Lin, T-H., Chang, R-X., A Secure Dynamic ID based Remote User Authentication Scheme for Multi-server Environment using Smart Cards, Expert Systems with Applications (2011), doi: 10.1016/j.eswa.2011.04.190] is analyzed. Three kinds of attacks are presented in different scenarios

***Key words**: Authentication, smart cards, dynamic ID, multi-server system, password, attack*

1. Introduction

Recently, Lee et al. gave six requirements for password authentication scheme for multi-server environment [1]. They also proposed a new scheme using smart cards for password authentication over insecure networks and claimed that it satisfied all the six requirements and thus is immune to various attacks. In this paper, however, some security loopholes of their scheme will be pointed out and the corresponding attacks will be described.

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

2. Lee et al.'s scheme

In this section, we will briefly review Lee et al.'s scheme. Their scheme consists of four phases: registration phase, login phase, verification phase, and password change phase. In order to facilitate future references, frequently used notations are listed below with their descriptions.

- U_i: The i th user;
- ID_i: The identity of U_i;
- PW_i: The password of U_i;
- S_j: The j th server;
- RC: The registration center;
- SC: A smart card;
- SID_j: The identity of S_j;
- CID_j: The dynamic ID of U_i;
- x, y: Two secret keys maintained by registration center;
- $h()$: A one-way hash function;
- $\oplus$: The bitwise XOR operation;
- $||$: String concatenation operation

Three entities: the user (U_i), the server (S_j), and the registration center (RC) are involved in Lee et al.'s scheme. First, RC chooses the master key x and secret number y to compute $h(x || y)$ and $h(y)$, and then shares them with S_j in the secure channel. Only RC knows the master secret key x and secret number y.

2.1. Registration phase

In this phase, everyone who wants to register at the server should submit his identity and password to RC and obtain a smart card. The detail of the phase is described as follows.

1) U_i generates a random number b_i, chooses his identity ID_i and PW_i, and computes $h(b_i \oplus PW_i)$. Then U_i sends ID_i and $h(b_i \oplus PW_i)$ to the registration center RC through a secure channel.

2) After receiving ID_i and $h(b_i \oplus PW_i)$, RC computes $T_i = h(ID_i || x)$, $V_i = T_i \oplus h(ID_i || h(b_i \oplus PW_i))$, $B_i = h(h(b_i \oplus PW_i) || H(x || y))$ and $H_i = h(T_i)$. Then RC stores $\{V_i, B_i, H_i, h(), h(y)\}$ into a smart card and issue it to U_i.

3) When receiving the smart card, U_i keys b_i into it and finish the registration.

2.2. Login phase

Once the user U_i wants to login to the server, as shown in Fig. 1, he will perform the following login steps.

1) U_i inserts his smart card into the smart card reader and then inputs ID_i and PW_i.

2) The smart card computes $T_i = V_i \oplus h(ID_i \| h(b_i \oplus PW_i))$ and $H_i' = h(T_i)$. If H_i' does not equal H_i, the smart card stops the request.

3) The smart card generates a random number N_i and computes $A_i = h(T_i \| h(y) \| N_i)$, $CID_i = h(b_i \oplus PW_i) \oplus h(T_i \| A_i \| N_i)$, $Q_i = h(B_i \| A_i \| N_i)$, and $P_{ij} = T_i \oplus h(h(y) \| N_i \| SID_j)$. Then, the smart card sends $M_1 = \{CID_i, P_{ij}, Q_i, N_i\}$ to the serer S_j.

2.3. Verification phase

This phase is executed by the server to determine whether the user is allowed to login or not. S_j executes the following steps to verify the legitimacy of U_i. We use Fig. 1 to demonstrate the phase.

1)Upon receiving M_1, S_j computes $T_i = P_{ij} \oplus h(h(y) \| N_i \| SID_j)$, $A_i = h(T_i \| h(y) \| N_i)$, $h(b_i \oplus PW_i) = CID_i \oplus h(T_i \| A_i \| N_i)$ and $B_i = h(h(b_i \oplus PW_i) \| h(x \| y))$. Then S_j computes $h(B_i \| A_i \| N_i)$ and checks it with Q_i. If they are not equal, S_j rejects the login request and terminates this session. Otherwise, S_j generates a random number N_j to compute $M_{ij}' = h(B_i \| N_i \| A_i \| SID_j)$. Finally, S_j sends the message $M_2 = \{M_{ij}', N_j\}$ to U_i.

2) Upon receiving M_2, U_i checks whether $h(B_i \| N_i \| A_i \| SID_j)$ equals M_{ij}'. If they are not equal, U_i stops the session. Otherwise, U_i computes $M_{ij}'' = h(B_i \| N_j \| A_i \| SID_j)$. At last, U_i sends $M_3 = \{M_{ij}''\}$ to S_j.

3)Upon receiving M_3, S_j checks whether $h(B_i \| N_j \| A_i \| SID_j)$ equals M_{ij}''. If they are not equal, U_i stops the request Otherwise, U_i is authenticated successfully.

After finishing verification phase, U_i and S_j can compute $SK = h(B_i \| N_i \| N_j \| A_i \| SID_j)$ as the session key for securing communications with authenticator. The login phase and verification phase are depicted in Figure 1.

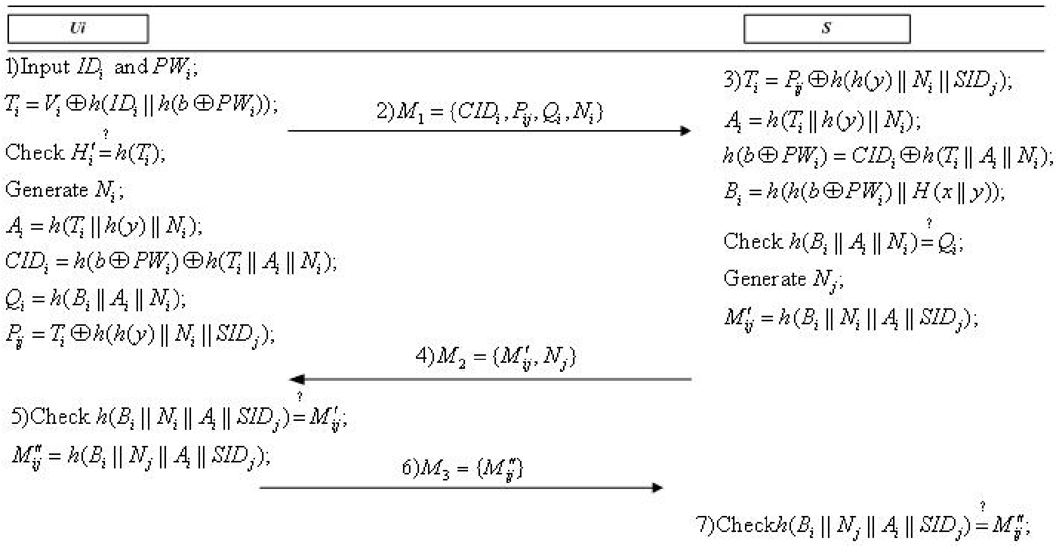

Fig. 1. Login phase and verification phase of Lee et al.'s scheme

2.4. Password change phase

This phase will be invoked if the client wants to change his password from PW_i to PW_{new}.

1) U_i inserts his smart card into the smart card reader and then inputs ID_i and PW_i.

2) The smart card computes $T_i = V_i \oplus h(ID_i \| h(b_i \oplus PW_i))$ and $H_i' = h(T_i)$. If H_i' does not equal H_i, the smart card stops the request.

3) U_i inputs the new password PW_{new} and a new random number b_{new}, computes $h(b_{new} \oplus PW_{new})$, $V_{new} = T_i \oplus h(ID_i \| h(b_{new} \oplus PW_{new}))$. At last, U_i sends ID_i and $h(b_{new} \oplus PW_{new})$ to RC through a secure channel.

4) Upon receiving ID_i and $h(b_{new} \oplus PW_{new})$, RC computes $B_{new} = h(h(b_{new} \oplus PW_{new}) \| h(x \| y))$ and sends it to U_i.

5) The smart card replaces V_i and B_i with V_{new} and B_{new}.

3. Cryptanalysis of Lee et al.'s scheme

In this section, we will demonstrate that Lee et al.'s scheme is vulnerable to impersonation attack, server spoofing attack, and can not provide two-factor security.

3.1 Impersonation attack

In this subsection, we first show that any malicious legal user can impersonate other legal users to log into remote server. Then we demonstrate that any malicious server also can impersonate any other legal users to log into remote server.

- **Malicious user's impersonation attack**

We assume that the adversary Z is a legal user of the system, and then he can obtain a smart card containing $\{V_Z$, B_Z, H_Z, $h()$, $h(y)$, $b_Z\}$. When another legal user U_i communicates with S_j, the adversary Z can intercept the login message $\{CID_i, P_{ij}, Q_i, N_i\}$ between U_i and S_j, and impersonate U_i though the following steps. We use Fig. 2 to demonstrate the attack.

1) Z two random numbers r and N', sets $T_i' \leftarrow r$, computes $A_i' = h(T_i' \| h(y) \| N_i')$, $CID_i' = h(b_Z \oplus PW_Z) \oplus h(T_i' \| A_i' \| N_i')$, $Q_i' = h(B_Z \| A_i' \| N_i')$, and $P_{ij}' = T_i' \oplus h(h(y) \| N_i' \| SID_j)$. Then, the smart card sends $M_1' = \{CID_i', P_{ij}', Q_i', N_i'\}$ to the serer S_j.

2)Upon receiving M_1', S_j computes $T_i = P_{ij}' \oplus h(h(y) \| N_i' \| SID_j) = T_i'$, $A_i = h(T_i \| h(y) \| N_i') = h(T_i' \| h(y) \| N_i') = A_i'$, $h(b_i \oplus PW_i) = CID_i' \oplus h(T_i \| A_i \| N_i') = h(b_Z \oplus PW_Z)$ and $B_i = h(h(b_i \oplus PW_i) \| h(x \| y)) = h(h(b_Z \oplus PW_Z) \| h(x \| y)) = B_Z$. It is obvious that $h(B_i \| A_i \| N_i')$ equals Q_i' since $Q_i' = h(B_Z \| A_i' \| N_i')$ and $B_i = B_Z$. Then, S_j generates a random number N_j to compute $M_{ij}' = h(B_i \| N_i \| A_i \| SID_j)$. Finally, S_j sends the message $M_2' = \{M_{ij}', N_j\}$ to Z.

3) Upon receiving M', Z computes $M_{ij}'' = h(B_Z \| N_j \| A_i' \| SID_j)$ and sends $M_3' = \{M_{ij}''\}$ to S_j.

4)Upon receiving M_3', S_j checks whether $h(B_i \| N_j \| A_i \| SID_j)$ equals M_{ij}''. It is obvious $h(B_i \| N_j \| A_i \| SID_j)$ equals M_{ij}'' since $B_i = B_Z$ and $M_{ij}'' = h(B_Z \| N_j \| A_i' \| SID_j)$. Then Z impersonate U_i successfully.

After above steps, Z and S_j can compute $SK = h(B_Z \| N_i' \| N_j \| A_i' \| SID_j)$ as the session key for securing communications with authenticator.

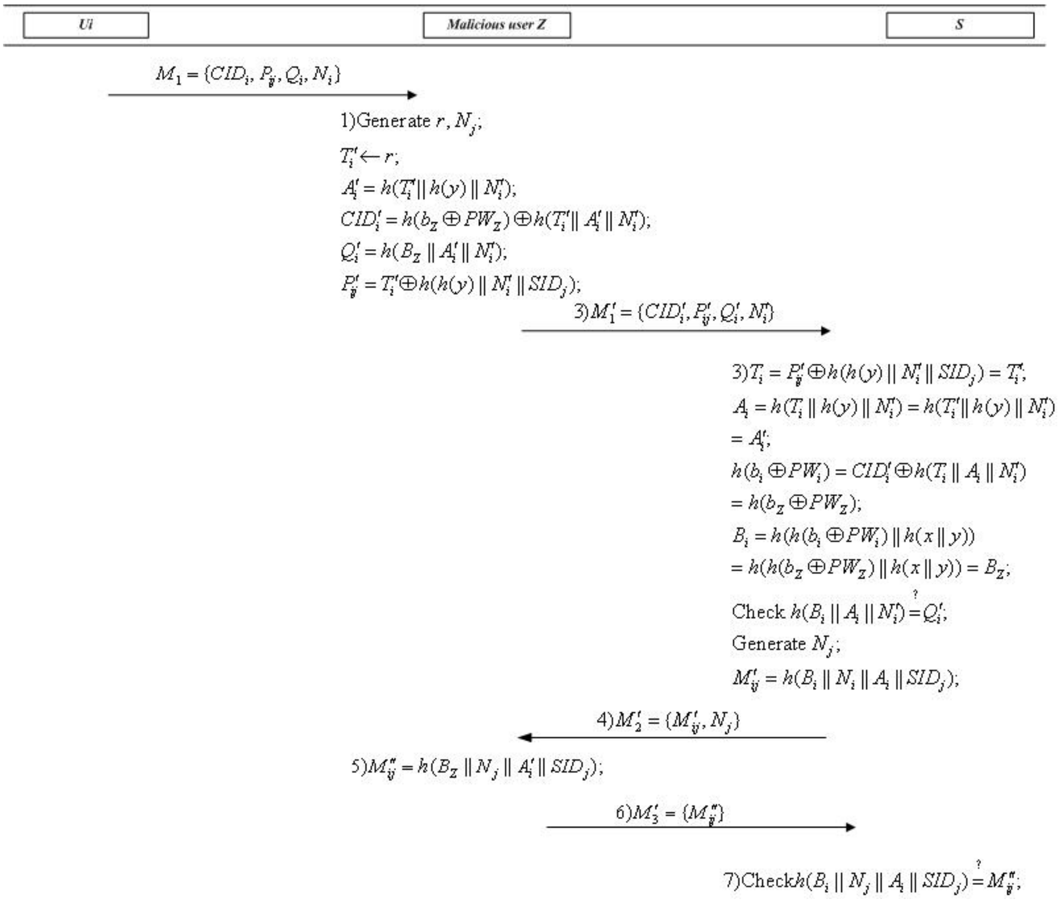

Fig. 2. Malicious user's impersonation attack

- **Malicious server's impersonation attack**

We assume that S_j is a malicious server of the system, and then he can obtain $h(x \| y)$ and $h(y)$ from RC. When a legal user U_i communicates with S_j, S_j can impersonate this user to obtain the services from other servers S_{j+1}. The detail of the attack, as shown in Fig. 3, is described as follows.

1) When receiving $M_1 = \{CID_i, P_{ij}, Q_i, N_i\}$ from U_i, S_j uses his $h(y)$ and $h(x \| y)$ to compute $T_i = P_{ij} \oplus h(h(y) \| N_i \| SID_j)$, $A_i = h(T_i \| h(y) \| N_i)$, $h(b_i \oplus PW_i) = CID_i \oplus h(T_i \| A_i \| N_i)$, $B_i = h(h(b_i \oplus PW_i) \| H(x \| y))$, and

$P_{ij+1} = T_i \oplus h(h(y) \| N_i \| SID_{j+1})$. Then S_j sends $M_1' = \{CID_i, P_{ij+1}, Q_i, N_i\}$ to another server S_{j+1}.

2) Upon receiving M_1 , S_{j+1} computes $T_i = P_{ij+1} \oplus h(h(y) \| N_i \| SID_{j+1})$, $A_i = h(T_i \| h(y) \| N_i)$, $h(b_i \oplus PW_i) = CID_i \oplus h(T_i \| A_i \| N_i)$ and $B_i = h(h(b_i \oplus PW_i) \| H(x \| y))$. Then S_{j+1} computes $h(B_i \| A_i \| N_i)$ and checks if it equals Q_i. It is obvious $h(B_i \| A_i \| N_i)$ equals Q_i. Then, S_{j+1} generates a random number N_{j+1} to compute $M_{ij+1}' = h(B_i \| N_i \| A_i \| SID_{j+1})$. Finally, S_{j+1} sends the message $M_2 = \{M_{ij+1}', N_{j+1}\}$ to S_j.

3) Upon receiving $\{M_{ij+1}', N_{j+1}\}$, S_j computes $M_{ij+1}'' = h(B_i \| N_{j+1} \| A_i \| SID_{j+1})$ and sends $M_3 = \{M_{ij+1}''\}$ to S_{j+1}.

4) Upon receiving M_3 , S_{j+1} checks whether $h(B_i \| N_{j+1} \| A_i \| SID_{j+1})$ equals M_{ij+1}'' . From the computation of M_{ij+1}'' we knows $h(B_i \| N_{j+1} \| A_i \| SID_{j+1})$ equals M_{ij+1}''. Then S_j impersonate U_i successfully.

After above steps, S_j and S_{j+1} can compute $SK = h(B_i \| N_i \| N_{j+1} \| A_i \| SID_{j+1})$ as the session key for securing communications with authenticator.

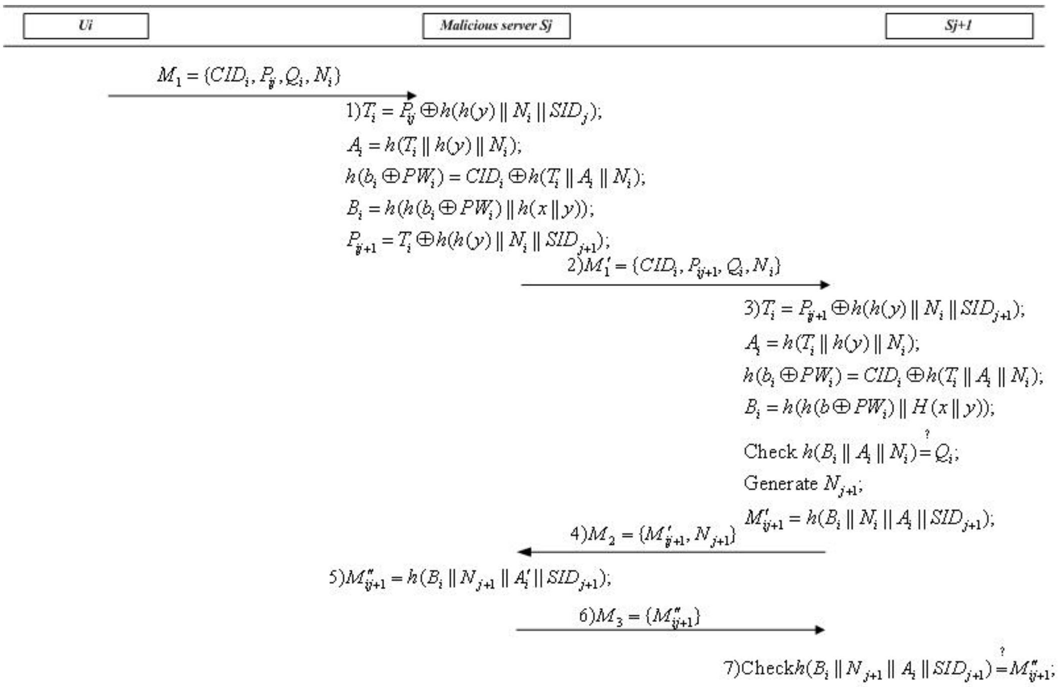

Fig. 3. Malicious server’s impersonation attack

3.2. Server spoofing attack

We assume that S_j is a malicious server of the system, and then he can obtain $h(x \| y)$ and $h(y)$ from RC. When another legal user U_i communicates with S_{j+1}, S_j can intercept the login message $M_1 = \{CID_i, P_{ij+1}, Q_i, N_i\}$ between U_i and S_{j+1}, and impersonate S_{j+1} though the following steps, where $CID_i = h(b_i \oplus PW_i) \oplus h(T_i \| A_i \| N_i)$, $Q_i = h(B_i \| A_i \| N_i)$, $P_{ij+1} = T_i \oplus h(h(y) \| N_i \| SID_{j+1})$ and $T_i = h(ID_i \| x)$. We use Fig. 4 to demonstrate the attack.

1)Upon receiving M_1, S_j computes $T_i = P_{ij+1} \oplus h(h(y) \| N_i \| SID_{j+1})$, $A_i = h(T_i \| h(y) \| N_i)$, $h(b_i \oplus PW_i) = CID_i \oplus h(T_i \| A_i \| N_i)$ and $B_i = h(h(b_i \oplus PW_i) \| H(x \| y))$. Then S_j computes $h(B_i \| A_i \| N_i)$ and checks it with Q_i. If they are not equal, S_j rejects the login request and terminates this session. Otherwise, S_j generates a random number N_{j+1} to compute $M'_{ij+1} = h(B_i \| N_i \| A_i \| SID_{j+1})$. Finally, S_j sends the message $M_2 = \{M'_{ij+1}, N_{j+1}\}$ to U_i.

2) Upon receiving M_2, U_i checks whether $h(B_i \| N_i \| A_i \| SID_{j+1})$ equals M'_{ij+1}. If they are not equal, U_i stops the session. Otherwise, U_i computes $M''_{ij+1} = h(B_i \| N_{j+1} \| A_i \| SID_{j+1})$. At last, U_i sends $M_3 = \{M''_{ij}\}$ to S_j.

After finishing verification phase, U_i and S_j can compute $SK = h(B_i \| N_i \| N_{j+1} \| A_i \| SID_{j+1})$ as the session key for securing communications with authenticator. Then S_j impersonate S_{j+1} successfully.

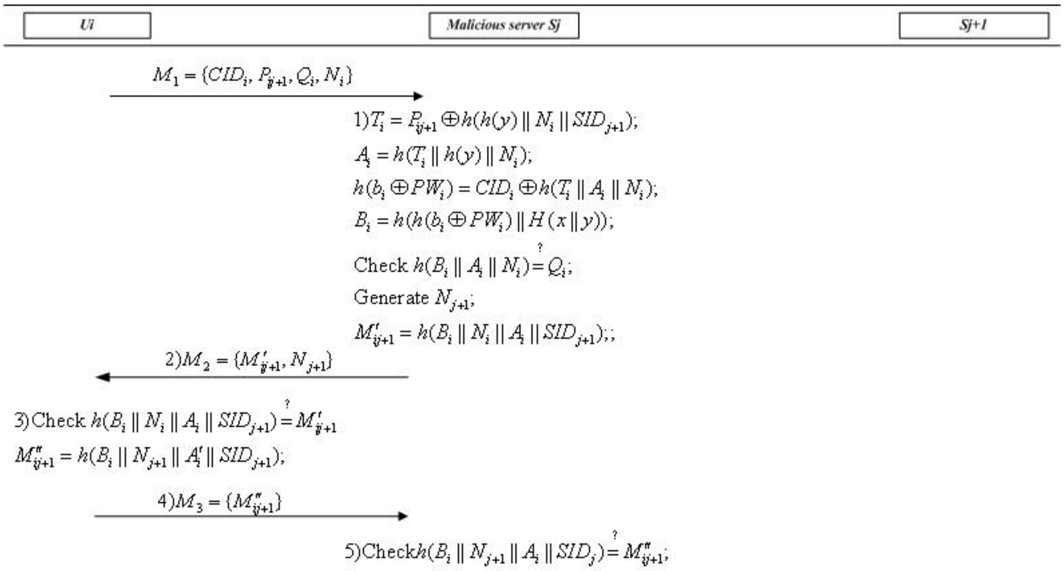

Fig. 4. Server spoofing attack

3.3. Password guessing attack

Although Sun et al. claim that their scheme can provide two-factor security, i.e. the user's password is secure even when the client's smart card is lost and the parameters in the card are derived[1], an password guessing attack will be given here.

Suppose the client's smart card is lost, an attacker A can read all the data, including $\{V_i$, B_i, H_i, $h()$, $h(y)$, $b\}$, from the smart card via physically access to the storage medium. He can get the password through the following steps.

1) A selects a password PW' from a uniformly distributed dictionary.
2) A computes $T_i = V_i \oplus h(ID_i \| h(b \oplus PW'))$ and $H_i' = h(T_i)$.
3) A check if H_i' equals H_i. If H_i' equals H_i, hen A find the correct passwords. Otherwise, A repeats steps 1, 2 and 3 until the correct password if found.

4. Conclusion

In [1], Lee et al. proposed a dynamic ID-based remote user authentication scheme for multi-server environment using smart cards and demonstrated its immunity against various attacks. However, after review of their scheme and analysis of its security, three kinds of attacks, i.e., impersonation attack, server spoofing attack, and password guessing attack, are presented in different scenarios. The analyses show that the scheme is insecure for practical application.

Reference

[1]. Lee, C-C., Lin, T-H., Chang, R-X., A Secure Dynamic ID based Remote User Authentication Scheme for Multi-server Environment using Smart Cards, Expert Systems with Applications (2011), doi: 10.1016/j.eswa.2011.04.190