LETTER

Cryptanalysis of a Dynamic ID-Based Remote User Authentication Scheme with Access Control for Multi-Server Environments

Debiao HE†, Nonmember and Hao HU†(a), Student Member

SUMMARY Recently, Shao et al. [M. Shao and Y. Chin, A privacy-preserving dynamic id-based remote user authentication scheme with access control for multi-server environment, IEICE Transactions on Information and Systems, vol. E95-D, no. 1, pp. 161–168, 2012] proposed a dynamic ID-based remote user authentication scheme with access control for multi-server environments. They claimed that their scheme could withstand various attacks and provide anonymity. However, in this letter, we will point out that Shao et al.’s scheme has practical pitfalls and is not feasible for real-life implementation. We identify that their scheme is vulnerable to two kinds of attacks and cannot provide anonymity.

key words: authentication scheme, multi-server environment, dynamic ID-based, anonymity

1. Introduction

With the rapid growth of Internet technologies, the demand for Internet services explodes accordingly. Multi-server architecture has been widely used in our life since a single server cannot provide enough services. To guarantee secure communication in multi-server environments, authentication schemes have been widely studied.

Very recently, Shao et al. [1] proposed a dynamic ID-based remote user authentication scheme with access control for multi-server environment and claimed that it 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 rest of this paper is structured as follows. In Sect. 2, we give a brief review of Shao et al.’s scheme. Section 3 presents our attacks on Shao et al.’s scheme. Section 4 concludes the paper.

2. Review of Shao et al.’s Scheme

For convenience, notations used throughout the paper are summarized as follows:

- $U_i$: the $i$th user.
- $ID_i$: the identity of $U_i$.
- $PW_i$: the password of $U_i$.
- $S_j$: the $j$th service provider.
- $SID_j$: the identity of $S_j$.
- $RC$: the registration center.
- $x$: the secret key of $RC$.
- $h(\cdot)$: a secure hash function.
- $\oplus$: the bitwise XOR operation.
- $\|$: the concatenation operation.

There are four phases in Shao et al.’s scheme: registration phase, login phase, authentication phase, password change phase and track phase.

2.1 Registration Phase

There are two sub-phase in this phase. Both of them are described as follows.

Server registration: The service provider $S_j$ submits his identity $SID_j$ to the registration center $RC$ via secure channel. $RC$ computes a secret number $y_j = h(h(x)||SID_j)$ and sends it to $S_j$ via secure channel. $S_j$ keeps $y_j$ secretly after receiving it.

User registration: Before a user $U_i$ can access services, he must register in $RC$ through the following steps.

1) $U_i$ submits his identity $ID_i$ and password $PW_i$ to $RC$ via secure channel.

2) After receiving $ID_i$ and $PW_i$, $RC$ generates the curve polynomial $F(L)[2]$ as access rights of $U_i$ and computes $T_i = h(ID_i||x_i), R_i = T_i \oplus h(ID_i) \oplus h(PW_i), V_i = T_i \oplus h(ID_i) \oplus h(PW_i)$, and $H_i = h(T_i)$. Then, $RC$ stores $\{R_i, V_i, H_i, F(L), h(\cdot)\}$ into a smart card and issues it to $U_i$ via secure channel.

2.2 Login Phase

When $U_i$ wants to access $S_j$, the following steps will be executed to verify $U_i$’s legality.

1) $U_i$ inserts his smart card into a card reader and inputs $ID_i^*$ and $PW_i^*$.

2) After receiving $ID_i^*$ and $PW_i^*$, $RC$ generates a random number $N_i$ and computes $\gamma_i^* = h((R_i \oplus T_i^* \oplus h(PW_i^*)) || SID_j^*)$, $CID_i = ID_i^* \oplus h((R_i \oplus T_i^* \oplus h(PW_i^*)) || N_i)$ and $Q_i = h(T_i^* || N_i)$.

3) The smart card generates a random number $N_i$ and computes $\gamma_i = h((R_i \oplus T_i \oplus h(PW_i)) || SID_j), CID_i = ID_i \oplus h((R_i \oplus T_i \oplus h(PW_i)) || N_i)$ and $Q_i = h(T_i || N_i)$.

4) The smart card uses $SID_j$ into polynomial $F(L)$ to get the role value $P$ and computes $P_{sk} = P \oplus h(\gamma_i || N_i), G_i = CID_i \oplus h(\gamma_i || N_i)$ and $C_i = h(CID_i || Q_i || P || N_i)$.

5) The smart card sends the login request message $\{C_i, G_i, Q_i, P_{sk}, N_i\}$ to $S_j$.

2.3 Authentication Phase

Upon receiving the login request message, $U_i$ and $S_j$ will...
carry out the following steps to authenticate each other.

1) \( S_j \) computes \( CID'_j = G_i \oplus h(y_j || N_j) \) and \( P'_j = P_i \oplus h(y_j || N_i) \). Then \( S_j \) checks whether \( C_i \) and \( h(CID'_j || Q_i || P'_j || N_i) \) are equal. If they are not equal, \( S_j \) stops the session.

2) \( S_j \) verifies \( Q_i \) with every data \( h(T_i || N_i) \) stored in the blacklist of malicious users given by \( RC \) at track phase. When no match is found, \( S_j \) rejects the login request.

3) \( S_j \) generates a random number \( N_j \) and computes \( M_1 = h(CID'_j || SID_j || N_j) \). At last, \( S_j \) sends \( \{ M_1, N_j \} \) to \( U_i \).

Upon receiving the message \( \{ M_1, N_j \} \), \( U_i \)'s smart card performs the following steps.

4) \( U_i \)'s smart card checks whether \( M_1 \) and \( h(CID'_i || SID_j || N_i) \) are equal. If they are not equal, \( U_i \)'s smart card stops the session.

5) \( U_i \)'s smart card computes \( M_2 = h(CID'_i || SID_j || N_j) \) and sends \( \{ M_2 \} \) to \( S_j \).

Upon receiving the message \( \{ M_2 \} \), \( S_j \)'s smart card performs the following steps.

6) \( S_j \)'s smart card checks whether \( M_2 \) and \( h(CID'_i || SID_j || N_j) \) are equal. If they are not equal, \( S_j \) stops the session. Otherwise, \( U_i \) is authenticated.

### 2.4 Password Change Phase

They could performs the following steps to change his password without the help of \( RC \).

1) \( U_i \) inserts his smart card into a card reader and inputs \( ID'_i \) and \( PW'_i \).

2) \( U_i \)'s smart card computes \( T'_i = V_i \oplus h(ID'_i || PW'_i) \). Then the smart card checks whether \( H_i \) and \( h(T'_i) \) are equal. If they are not equal, the smart card rejects the session.

3) \( U_i \) inputs the new password \( PW_{new} \) into the smart card. Then \( U_i \)'s smart card computes \( V'_i = T'_i \oplus h(ID'_i || PW_{new}) \) and \( R_{new} = R_i \oplus h(PW_{new}) \). Then \( U_i \)'s smart card computes replace \( V_i \) and \( R_i \) with \( V'_i \) and \( R_{new} \) respectively.

### 2.5 Track Phase

On discovering a malicious user \( U_i \), \( S_j \) collects the relevant data regarding \( U_i \) and obtains \( U_i \)'s real identity with the cooperation of \( RC \).

1) \( S_j \) sends \( CID_i \) and \( N_i \) to \( RC \).

2) Upon receiving \( CID_i \) and \( N_i \), \( RC \) computes \( ID'_i = CID_i \oplus h(h(x) || N_i) \) and \( T'_i = h(ID'_i || x) \).

3) \( RC \) updates the blacklist of malicious users with \( T'_i \) and sends the latest version of the blacklist to all of servers.

### 3. Cryptanalysis of Shao et al.'s Scheme

Since all the message are transmitted in the public network, then we could assume that an adversary \( A \) completely control the communication channel between \( U_i \) and \( S_j \), which means that he can insert, delete, or alter any messages in the channel. Besides, Kocher et al. [3] and Messerges et al. [4] have pointed out that all existent smart cards are vulnerable in that the confidential information stored in the device could be extracted by physically monitoring its power consumption. Then we could assume that all secrets in a card may be revealed once it is lost.

Shao et al. claimed that their scheme could resist various attacks. Basing on the above assumptions, we will demonstrate that Shao et al.'s scheme cannot withstand offline password guessing attack and stolen smart card attack.

#### 3.1 Server Spoofing Attack

Let \( A \) be a malicious user. He could get a legal smart card from \( RC \). Then \( R_A = T_A \oplus h(x) \oplus h(PW_A) \), \( V_A = T_A \oplus h(ID_A || PW_A) \) and \( H_A = h(T_A) \) are stored in his smart card, where \( T_A = h(ID_A || x) \). \( A \) could carry out the sever spoofing attack through the following steps.

1) \( A \) extracts \( R_A, V_A \) and \( H_A \) from his smart card.

2) \( A \) computes \( T_A = V_A \oplus h(ID_A || PW_A) \), \( h(x) = R_A \oplus T_A \oplus h(PW_A) \) and \( S_j \)'s secret key \( y_j = h(h(x) || SID_i) \).

3) \( A \) intercepts the message \( \{ C_i, G_i, Q_i, P_i, N_i \} \) sent by \( U_i \).

4) \( A \) generates a response message \( \{ M_1, N_j \} \) as \( S_j \) does in Sect. 2.3 since \( A \) knows \( S_j \)'s secret key. Then \( A \) sends \( \{ M_1, N_j \} \) to \( U_i \).

It is easy to say that the message \( \{ M_1, N_j \} \) could pass the verification of \( U_i \) since it is generated by \( S_j \)'s secret key \( y_j = h(h(x) || SID_i) \). Therefore, Shao et al.'s scheme is vulnerable to the server spoofing attack.

#### 3.2 Password Guessing Attack

Let \( U_i \) be a legal user. We assume that \( U_i \)'s smart card is lost and gotten by an adversary \( A \). Then \( A \) could extract the confidential information \( R_i = T_i \oplus h(x) \oplus h(PW_i) \), \( V_i = T_i \oplus h(ID_i || PW_i) \) and \( H_i = h(T_i) \), stored in his smart card, where \( T_i = h(ID_i || x) \). Using \( R_i, V_i \) and \( H_i \), \( A \) could get \( U_i \)'s password \( PW_i \) through the following steps.

1) \( A \) guesses a password \( PW'_i \) and an identity \( ID'_i \).

2) \( A \) computes \( T'_i = V'_i \oplus h(ID'_i || PW'_i) \).

3) \( A \) checks whether \( H_i \) and \( h(T'_i) \) are equal. If they are equal, \( A \) finds the correct password. Otherwise, \( A \) repeats steps 1), 2) and 3) until the correct password is found.

In Shao et al.’s scheme, users could select their passwords and identities freely. For convenience, they would like to choose human-memorable short strings as passwords and identities. Then, the size of two corresponding dictionaries of passwords and identities is very small. Therefore, our attack is feasible because both password and identity are not high-entry keys. Besides, the attacker can probably deduce the user’s identity when she gets the smart card. In that case, our attack can be done much more efficiently since she only needs to guess the password. This assumption is reasonable because the user often choose his name as his identity or write his identity on the card; and moreover the input identity is usually displayed in plain on the screen and thus can be possibly seen when the attacker steals the card [5]. Therefore, Shao et al.’s scheme is vulnerable to the password guessing attack.
3.3 Inability of Providing Anonymity

Shao et al. claimed that their scheme could provide anonymity. However, we will show a malicious user could get the real identity of a user who sends the login request message. Let $A$ be a malicious user. He could get a legal smart card from $RC$. Then $R_A = T_A \oplus h(x) \oplus h(PW_A)$, $V_A = T_A \oplus h(ID_A||PW_A)$, and $H_A = h(T_A)$ are stored in his smart card, where $T_A = h(ID_A||x)$. $A$ could get the real identity through the following steps.

1) $A$ extracts $R_A$, $V_A$ and $H_A$ from his smart card.

2) $A$ computes $T_A = V_A \oplus h(ID_A||PW_A)$, $h(x) = R_A \oplus T_A \oplus h(PW_A)$ and $S_j$’s secret key $y_j = h(h(x)||SID_j)$.

3) Upon intercepting the login request message $\{C_i, G_i, Q_i, P_L, N_i\}$, $A$ computes $CID_i = G_i \oplus h(y_j||N_i)$ and $ID_i = CID_i \oplus h(h(x)||N_i)$.

It is easy to say that $A$ could get the real identity of the user, who sends the login request message $\{C_i, G_i, Q_i, P_L, N_i\}$. Therefore, Shao et al.’s scheme could not provide anonymity.

4. Conclusion

Recently, Shao et al. [1] proposed a dynamic ID-based remote user authentication scheme with access control for multi-server environment and demonstrated its immunity against various attacks. However, after reviewing their scheme and analyzing its security, two kinds of attacks, i.e., server spoofing attack and password guessing attack, are presented in different scenarios. Moreover, we also demonstrate that Shao et al.’s scheme could not provide anonymity. The analysis shows that their scheme is insecure for practical applications.

Acknowledgements

The authors thank Dr. Yuji Watanabe and the anonymous reviewers for their valuable comments. This research was supported by the Specialized Research Fund for the Doctoral Program of Higher Education of China (No. 20110141120003).

References