On the Security of Baruah et al.’s
Biometric-based Multi-server Authentication
Scheme Using Smart Card

Eun-Jun Yoon¹

Department of Cyber Security, Kyungil University
33 Buho-Ri, Hayang-Ub, Kyungsan-Si
Kyungsangpuk-Do 712-701, Republic of Korea

Abstract
In 2015, Baruah et al. proposed a biometric-based multi-server authentication scheme using smart card. Baruah et al. claimed that their proposed scheme is secure to various cryptographic attacks and can provide forward secrecy. However, this paper points out that Baruah et al.’s scheme not only is still vulnerable to identity guessing attack using stolen smart card and replay attack, but also cannot provide perfect forward secrecy unlike their claims. For this reason, Baruah et al.’s scheme is insecure for practical application.

Keywords: Cryptography; Biometric authentication; Smart cards; Cryptanalysis; Guessing attack; Replay attack; Perfect forward secrecy

1 Introduction

Recently, biometric-based multi-server authentication scheme using smart card is a widely used and researched method because it has resolved the problem of users to manage the different identities and passwords[1, 2]. In 2014, Mishra et al.[1] proposed an improved multi-server based authentication scheme using smart cards. Unfortunately, Baruah et al.[2] pointed out that Mishra et al.’s

¹Corresponding author. Tel.: +82-53-600-5623; Fax: +82-53-600-5639
scheme is insecure against stolen smart card attack and impersonation attacks. Baruah et al. also proposed a security enhanced authentication scheme to eliminate all identified weaknesses of Mishra et al.’s scheme.

Nevertheless, this paper points out that Baruah et al.’s scheme has the following security problems[3, 4, 5, 6] unlike their claims: (1) The scheme is still vulnerable to identity guessing attack with stolen smart card[3]. (2) The scheme is vulnerable to replay attack[4]. (3) The scheme cannot provide perfect forward secrecy[5, 6]. For this reason, Baruah et al.’s scheme is insecure for practical application.

This paper is organized as follows: Section 2 briefly reviews the Baruah et al.’s biometric-based multi-server authentication scheme using smart card. The security flaws of Baruah et al.’s scheme are shown in Section 3. Finally, conclusions are given in Section 4.

2 Review of Baruah et al.’s Scheme

This section briefly reviews Baruah et al.’s biometric-based multi-server authentication scheme using smart card[2]. There are four phases in the Baruah et al.’s scheme: registration phase, login phase, authentication phase and password change phase. We outlined some notations used in this research paper.

- $ID_i$: Identity of the $i^{th}$ user
- $SID_j$: Identity of the $j^{th}$ server
- $PW_i$: Password of the $i^{th}$ user
- $BIO_i$: Biometric information of the $i^{th}$ user
- $PSK$: Pre-shared secret key of the servers
- $x$: Master secret key maintained by the registration center
- $T_r$: Registration Time of the user
- $h(\cdot)$: A secure one-way hash function
- $N_{i_1}, n_1$: Random nonce of the $i^{th}$ user
- $N_{j_2}, n_2$: Random nonce of the $j^{th}$ server
- $\oplus$: Bit-wise Exclusive-OR operation
- $\|$: Message concatenation operation

2.1 Registration Phase

In the registration phase, the registration center provides secrets to the user as well as the server. Basically, it can be sub-categorized into the server registration phase and the user registration phase.
2.1.1 Server Registration Phase

When a server wants to provide some service to the public, then it has to first register itself to the registration center.

1. Server → Registration center: $SID_j$
   The server sends a join request along with its identity $SID_j$ to the registration center.

2. Registration center → Server: \[ h(SID_j || h(PSK)), h(PSK || x) \]
   The registration center replies with $h(SID_j || h(PSK))$ and $h(PSK || x)$ through the Internet Key Exchange Protocol version 2 (IKEv2) [2].

3. The server uses these secret \( \{ h(SID_j || h(PSK)), h(PSK || x) \} \) to authenticate any registered user.

2.1.2 User Registration Phase

The users must first register themselves if they want to access any services provided by the set of registered servers.

1. User → Registration center: \{ $ID_i, R_1$ \}
   The user submits his/her identity $ID_i$ and $R_1 = h(PW_i || BIO_i)$ through a secure channel.

2. Registration center → User: Smart card \( SC_i = \{ B_i, C_i, D_i, E_i, h(\cdot) \} \)
   The registration center then computes the following:
   \[ A_i = h(ID_i || x) \]
   \[ B_i = h(PSK || x) \oplus A_i \]
   \[ C_i = h(R_i || ID_i) \oplus h(A_i) \]
   \[ D_i = h(PSK) \oplus h(ID_i) \]
   \[ E_i = R_1 \oplus ID_i \]
   The registration center creates a smart card \( SC_i = \{ B_i, C_i, D_i, E_i, h(\cdot) \} \). This personalized smart card is provided to the user via a secure channel.
2.2 Login Phase

Fig. 1 depicts the Baruah et al.’s user login and authentication phases. The login phase performs as follows:

1. The user inserts the smart card and then inputs his/her identity $ID_i$, password $PW_i$ and biometric information $BIO_i$.

2. The smart card computes $R_1 = h(PW_i\|BIO_i)$ and extracts $ID_i = R_1 \oplus E_i$ by using the stored identity $E_i = R_1 \oplus ID_i$. The smart card verifies whether the entered identity $ID_i$ is equal to stored identity $ID_i = R_1 \oplus E_i$ or not. If failure occurs, the login phase is immediately aborted. Otherwise, proceeds for the succeeding steps.

3. The smart card extracts the following messages from the stored data.

   \[
   h(PSK) = h(ID_i) \oplus D_i \tag{6}
   \]

   \[
   h(A_i) = C_i \oplus h(R_1\|ID_i) \tag{7}
   \]

4. The smart card randomly generates a nonce $N_i$ and then computes the following messages:

   \[
   M_1 = h(SID_j\|h(PSK)) \oplus h(ID_i\|N_i) \tag{8}
   \]

   \[
   M_2 = N_i \oplus h(A_i) \tag{9}
   \]

   \[
   V_1 = h(N_i \oplus B_i) \tag{10}
   \]

5. Smart card $\rightarrow$ Server: \{$B_i, M_1, M_2, V_1$\} The smart card transmits the message \{$B_i, M_1, M_2, V_1$\} to the server $SID_j$ via a public channel for authentication.

2.3 Authentication Phase

Upon receiving the authentication messages, the server $SID_j$ performs the following set of operations to agree on the same session key.

1. The server computes the following messages by using its secrets:

   \[
   A_i = B_i \oplus (PSK|x) \tag{11}
   \]

   \[
   h(ID_i\|N_i) = M_1 \oplus h(SID_j\|h(PSK)) \tag{12}
   \]

   \[
   N_i = M_2 \oplus h(A_i) \tag{13}
   \]
2. The server verifies whether $V_1$ is equal to the computed value $h(N_i \oplus B_i)$ or not. If this holds, then the server generates a random nonce $N_j$. On failure, the phase is simply exited.

3. The server uses the user’s information and its nonce $N_j$ and identity $SID_j$ to generate the following session key:

$$SK_{ji} = h(h(ID_i||N_i)||SID_j||B_i||N_j)$$

(14)

4. The server sends the following messages to the user via a public channel:

$$M_3 = N_j \oplus h(ID_i||N_i)$$

(15)

$$V_2 = N_i \oplus h(SK_{ji}||N_j)$$

(16)

5. The user computes $N_j$ from $M_3$. It then uses the information to compute the following session key:

$$SK_{ij} = h(h(ID_i||N_i)||SID_j||B_i||N_j)$$

(17)

6. The user verifies whether $N_i$ is equal to the computed value $V_2 \oplus h(SK_{ij}||N_j)$ or not. If this holds, then the user authenticates the corresponding party is legal server.

2.4 Password Change Phase

The password change phase performs as follows:

1. When $U_i$ wants to change the password, the user inserts his/her smart card into the machine and inputs his/her identity $ID_i$, password $PW_i$ and biometric $BIO_i$.

2. The smart card checks the entered information. If the user is the authentic one, then the card prompts the user for new password $PW_i^*$ and computes the followings:

$$R_i^* = h(PW_i^*||BIO_i)$$

(18)

$$E_i^* = E_i \oplus R_1 \oplus R_i^*$$

(19)

$$C_i^* = h(R_i^*||ID_i) \oplus h(R_1||ID_i) \oplus C_i$$

(20)

3. The smart card updates $E_i^*$ and $C_i^*$ in the place of $E_i$ and $C_i$. Now, the updated smart card has $SC_i = \{B_i, C_i^*, D_i, E_i^*, h(\cdot)\}$. 

3 Cryptanalysis of Baruah et al.’s Scheme

This section demonstrates that Baruah et al.’s scheme[2] not only cannot withstand an identity guessing attack with stolen smart card[3] and replay attack[4], but also cannot provide perfect forward secrecy[6].

3.1 Identity Guessing Attack

An identity guessing attack is the most powerful attack to the password-based authentication schemes. In the identity guessing attack, an attacker uses a guessed identity to verify the correctness of the identity in an off-line manner. The attacker can freely guess an identity and then check if it is correct without limitation in the number of guesses[3].

Suppose that the attacker steals the parameters $SC_i = \{B_i, C_i, D_i, E_i, h(.)\}$ from a lost smart card and knows the server’s response message $\{B_i, M_1, M_2, V_1\}$.

Then the attacker can perform an identity guessing attack to obtain the identity $ID_i$ of the user as follows:

1. The attacker selects a candidate identity $ID_i^*$. 

---

**Figure 1: Login and Authentication Phases**
2. The attacker checks if the following equation holds or not
\[
V_1 = h(M_2 \oplus C_i \oplus h(E_i \oplus ID_i^* || ID_i^*) \oplus B_i)
\] (21)

If the check passes, then the attacker confirms that the guessed identity \(ID_i^*\) is the correct one.

3. If it is not correct, the attacker chooses another identity \(ID_i^{**}\) and repeatedly performs above steps (2) \sim (5) until
\[
V_1 = h(M_2 \oplus C_i \oplus h(E_i \oplus ID_i^{**} || ID_i^{**}) \oplus B_i)
\] (22)

It is clear that if \(ID_i^* = ID_i\), then
\[
\begin{align*}
&h(M_2 \oplus C_i \oplus h(E_i \oplus ID_i^* || ID_i^*) \oplus B_i) \\
&= h(N_i \oplus h(A_i) \oplus h(R_i || ID_i) \oplus h(A_i) \oplus h(R_i \oplus ID_i \oplus ID_i^* || ID_i^*) \oplus B_i) \\
&= h(N_i \oplus h(R_i || ID_i) \oplus h(R_i || ID_i^*) \oplus B_i) \\
&= h(N_i \oplus B_i) \\
&= V_1
\end{align*}
\] (23)

Therefore, Baruah et al.’s scheme is vulnerable to the identity guessing attack. The algorithm of the identity guessing attack for getting the identity \(ID_i^*\) is as follows:

**Identity Guessing Attack**\((SC_i = \{B_i, C_i, D_i, E_i, h(\cdot)\}, M_2, V_1, D_ID)\) \{
for \(i := 0\) to \(|D_ID|\) \{
\begin{align*}
&ID_i^* \leftarrow D_ID; \\
&R_i^* = E_i \oplus ID_i^*; \\
&h(A_i^*) = C_i \oplus h(R_i^* || ID_i^*); \\
&N_i^* = M_2 \oplus h(A_i^*); \\
\text{if } V_1 \overset{?}{=} h(N_i^* \oplus B_i) \text{ then} \\
&\text{return } ID_i^*
\end{align*}
\}
\}

The running time of the above identity guessing attack is \((O(|D_ID|) \times 4T_X \times 3T_H)\), where \(T_X\) and \(T_H\) represent the execution time of bit-wise XOR operations and hash operations respectively. The search spaces \(D_ID\) is unlikely to be large enough (for example, \(|D_ID| \leq 10^6\), and the time complexities \(T_X\) and \(T_H\) all can be executed with negligible amount of time, thus the polynomial time-bounded attacker can find the exact identity \(ID\) of the user easily\[3\].


3.2 Replay Attack

Baruah et al.’s scheme is insecure to replay attack which an attacker eavesdrops a communication between a user and the server and try to use these messages for opening a communication to a server in future. Replay attack refers to retransmission of previously intercepted invalid message. In password authentication schemes, replay attack is usually concerned in the case of a illegal client attempts to impersonate a legal one by replaying previously intercepted invalid message to the server[4]. An attacker can performs the replay attack as follows:

1. Attacker eavesdrops a communication and stores the login messages \{M_1, M_2, V_1, B_i\} for performing replay attack in future, where

\[
M_1 = h(SID_j || h(PSK)) \oplus h(ID_i || N_i) \quad (24)
\]

\[
M_2 = N_i \oplus h(A_i) \quad (25)
\]

\[
V_1 = h(N_i \oplus B_i) \quad (26)
\]

2. Attacker transmits these stored messages \{M_1, M_2, V_1, B_i\} to a registered server \(SID_j\).

3. Upon receiving the messages \{M_1, M_2, V_1, B_i\} from the attacker, the server \(SID_j\) will retrieve the followings:

\[
A_i = h(PSK||x) \oplus B_i \quad (27)
\]

\[
h(ID_i || N_i) = h(SID_j || h(PSK)) \oplus M_1 \quad (28)
\]

\[
N_i = M_2 \oplus h(A_i) \quad (29)
\]

4. The server \(SID_j\) will verify whether \(V_1\) is equal to the computed value \(h(N_i \oplus B_i)\) or not. We can see that this verification always holds because the received messages has not been modified by the attacker.

5. Upon verification, the server \(SID_j\) will select a random nonce \(N_j^*\) and then generates the session key as

\[
SK_{ij}^* = h(h(ID_i||N_i)||SID_j||B_i||N_j^*) \quad (30)
\]

6. Finally, the server \(SID_j\) will transmit the following response messages by using the session key \(SK_{ij}^*\) to the attacker:

\[
M_3^* = N_j^* \oplus h(ID_i||N_i) \quad (31)
\]

\[
V_2^* = N_1 \oplus h(SK_{ij}||N_j^*) \quad (32)
\]

7. When receiving the messages \{\(M_3^*, V_2^*\}\) from the server \(SID_j\), the attacker drops the messages without performing the authentication step. As a result, the attacker will pass the authentication and is considered as a legal user in the authentication phase.
3.3 Perfect Forward Secrecy Problem

Perfect forward secrecy is a very important security requirement in evaluating a strong authentication scheme. A scheme with perfect forward secrecy assures that even if one entity’s long-term key is compromised, it will never reveal any session keys used before. For example, the well-known Diffie-Hellman key agreement scheme[5] can provide perfect forward secrecy. Baruah et al.’s authentication scheme, however, does not provide it because once the long term secret $A_i$ of the user and the pre-shared key $PSK$ of the server are disclosed, all previous fresh session keys $SK$ will also be opened and hence previous communication messages will be learned[6].

In Baruah et al.’s scheme, suppose an attacker obtains $A_i$ from the compromised user and $PSK$ from the compromised server, and intercepts transmitted values $\{M_1, B_i, SID_j, M_2\}$, then the attacker can compute $h(SID_j||h(PSK))$ and extract the user’s random value $h(ID_i||N_i)$ by computing $h(ID_i||N_i) = M_1 \oplus h(ID_i||N_i)$. The attacker can also extract the server’s random nonce $N_j$ by computing $N_j = M_3 \oplus h(ID_i||N_i)$. Finally, the attacker can compute the common shared session key $SK_{ij} = h(h(ID_i||N_i)||SID_j||B_i||N_j)$ between user and server. Therefore, the attacker can get the session key $SK_{ij} = SK_{ji}$. Obviously, Baruah et al.’s scheme does not provide perfect forward secrecy.

4 Conclusions

This paper reviewed Baruah et al.’s biometric-based multi-server authentication scheme using smart card and then pointed out that Baruah et al.’s scheme is still vulnerable to identity guessing attack using stolen smart card and replay attack unlike their claims and it cannot provide perfect forward secrecy. For this reason, Baruah et al.’s scheme is insecure for practical application. Further works will be focused on improving the Baruah et al.’s scheme which can be able to provide greater security and to be more efficient than the existing biometric-based multi-server authentication scheme using smart card by an accurate performance analysis.

Acknowledgements. This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP)(No. 2015R1A2A2A01006824).

References


Received: November 3, 2015; Published: December 18, 2015