Cryptanalysis of Robust Smart Card Secured Authentication Scheme on SIP Using Elliptic Curve Cryptography

Eun-Jun Yoon

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

Copyright © 2015 Eun-Jun Yoon. This article is distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

In 2014, Yeh et al. proposed a robust smart card secured authentication scheme on SIP using elliptic curve cryptography to conquer many forms of attacks in previous protocols. Yeh et al. claimed that their proposed protocol is more efficient than Diffie-Hellman’s concept authentication protocols for SIP and performs secured mutual authentication, which can be implemented on different real network environments such as VoIP. However, this paper points out that Yeh et al.’s protocol not only suffers from stolen smart card attack, but also does not provide perfect forward secrecy.

Keywords: Cryptography; Smart card authentication; Session initiation protocol; Elliptic Curve Cryptography; Stolen smart card attack; Perfect forward secrecy

1 Introduction

Session Initiation Protocol(SIP) has been widely used in current Internet protocols such as Hyper Text Transport Protocol(HTTP) and Simple Mail Transport Protocol(SMTP)[1]. SIP is a powerful signaling protocol that controls
communications on the Internet for establishing, maintaining and terminating sessions. However, the original authentication scheme for SIP does not provide strong security because it works based on HTTP Digest authentication noted in RFC2617[2]. The services that are enabled by SIP are equally applicable to mobile and ubiquitous computing. For example, a user can register its locations with a SIP server and then it will know the availability and location of the user. In addition, the location could be home, work-place or in mobile[1, 2].

Since Elliptic Curve Cryptography(ECC) provides a smaller key size than any other cryptosystem and has faster computations than half of the other public key systems at the same security levels[3], ECC is suitable to be used for higher security authentication. Recently, various ECC-based SIP authentication protocols have been proposed to strengthen the security[4, 5, 6, 7, 8, 9].

In 2014, Yeh et al.[9] also proposed a robust smart card secured authentication scheme on SIP using ECC to conquer many forms of attacks in previous protocols. Yeh et al. claimed that their proposed protocol is more efficient than Diffie-Hellman’s concept authentication protocols for SIP and performs secured mutual authentication, which can be implemented on different real network environments such as VoIP[9]. However, this paper points out that Yeh et al.’s protocol not only suffers from stolen smart card attack, but also does not provide perfect forward secrecy[7, 8, 10, 11, 12, 13].

This paper is organized as follows: Section 2 briefly reviews the Yeh et al.’s ECC-based authentication protocol for SIP. The security flaws of Yeh et al.’s protocol are shown in Section 3. Finally, conclusions are given in Section 4.

2 Review of ECC-based Authentication Protocol for SIP

This section briefly reviews Yeh et al.’s protocol[9]. The Yeh et al.’s protocol consists of four phases: initiation phase, registration phase, mutual authentication phase, and password change phase. We outlined some notations used in this research paper.

- $qs$: private key of system
- $K_{IDA}$: secret key of the user
- $k$: shared session key between client and server
- $N_r$: random number
- $T$: timestamp
- $G_P$: cyclic group of prime order $n$ of $P$
- $P$: large prime generator of group
Cryptanalysis of robust smart card secured authentication scheme

- $n$: order of elliptic curve
- $h(\cdot)$: secure one-way hash function: $\{0, 1\}^* \to \{0, 1\}$
- $H_1(\cdot)$: secure one-way hash function $H_1: \{0, 1\}^* \to G_P$
- $H_2(\cdot)$: secure one-way hash function $H_2: \{0, 1\}^* \to Z_P^*$
- $H_3(\cdot)$: secure one-way hash function $H_3: \{0, 1\}^* \to Z_P^*$
- $\times$: scalar multiplication of elliptic curve
- $\oplus$: exclusive operation
- $||$: string concatenation operation

2.1 Initiation phase

In this phase, the user and the server set up several system parameters and formula for session key generation as follows:

I1. Both user and server choose an elliptic curve order $n$ over $E_p(a, b)$ generated by $P$, where $n$ is a large number for the security considerations.

I2. The server randomly selects $qs \in Z_P^*$ as the private key.

I3. The system parameters $\{N_r, h(\cdot), H_1(\cdot), H_2(\cdot), H_3(\cdot)\}$ are pre-stored in the smart card of user, where $N_r$ is a large random number.

2.2 Registration phase

In this phase, the user wants to register to the remote server and setup the secret codes into the smart card for the user. The following steps are executed:

R1. client $\rightarrow$ server: \([\text{username}, pw_y]\)

The user enters his/her username $id$ and password $pw_x$ to compute the $pw_y = h(pw_x \oplus N_r)$, and then submits $id$ and $pw_y$ to remote server.

R2. The server computes $K_{IDA} = qs \times H_1(id) \in G_p$ as user’s authentication key.

R3. The server computes $B_A = h(id \oplus pw_y)$ and $W_A = h(pw_y||id) \oplus K_{IDA}$.

R4. The server stores the secret parameters \(\{B_A, W_A, h(\cdot), H_1(\cdot), H_2(\cdot), H_3(\cdot)\}\) to a smart card and then issues the smart card to the user over a secure channel.

R5. The user stores $N_r$ in the smart card. As a result, the user stores secret parameters \(\{B_A, W_A, N_r, h(\cdot), H_1(\cdot), H_2(\cdot), H_3(\cdot)\}\) into the smart card.
2.3 Mutual authentication phase

Assume that user wants to communicate with the remote SIP server; he/she must enter username and password.

M1. client → server: REQUEST\{username, T_1, M_A, R^*_A\}

The user enters \(pw_x\) and then computes \(pw_y = h(pw_x \oplus N_r)\) and \(B'_A = h(id \oplus pw_y)\). Next, the user confirms whether the \(B'_A\) is equal to \(B_A\). If they are equal, the user computes \(V = h(pw_y||id)\) and \(K_{IDA} = W_A \oplus V\). In addition, the user chooses a random point \(R_A = (R^x_A, R^y_A) \in E_P(a, b)\), where \(R^x_A\) and \(R^y_A\) are the \(x\) and \(y\) coordinates of point \(R_A\). At the timestamp \(T_1\), the user computes \(t_1 = H_2(T_1)\), \(M_A = R_A + t_1 \times K_{IDA}\) and \(R^*_A = R^x_A \times P\). Finally, the user sends the REQUEST message to the remote server.

M2. server → client: CHALLENGE\{realm,, T_2, M_S, M_k\}

When the server receives REQUEST message, the server computes \(U_{IDA} = H_1(id)\), \(R'_A = M_A - qs \times t_1 \times U_{IDA}\), where \(t_1 = H_2(T_1)\), \(U_{IDA} = (U_x, U_y)\) and \(R'_A = (R'_x, R'_y)\). The server authenticates the identity of the user by checking whether \(R'_x \oplus P\) is equal to \(R^*_A\). If it holds, the server chooses a random point \(R_S = (R^x_S, R^y_S) \in E_P(a, b)\) and then computes \(M_S = R_S + t_2 \times qs \times U_{IDA}\), the common session key \(k = H_3(U_x||R^x_A||R^x_S)\) and \(M_k = (k + R^x_S) \times P\) at the timestamp \(T_2\). Finally, the server sends the CHALLENGE message to the user.

M3. client → server: RESPONSE\{realm, username, response\}

When the user receives CHALLENGE message, the user computes \(R'_S = M_S - t_2 \times K_{IDA}\) to obtain \(R'_S = (R'_x, R'_y)\) of server, where \(t_2 = H_2(T_2)\). In addition, \(U_{IDA} = (U_x, U_y)\) is also obtained. The user computes \(k^* = H_3(U^x||R^x_A||R^x_S)\) and \(M^*_k = (k^* + R^x_S) \times P\) and then verify whether \(M^*_k = M_k\). If it holds, the server is authenticated by user; otherwise, terminate the process. Finally, the user computes \(response = h(\text{username}||\text{realm}||k)\) and then sends the RESPONSE message to the server.

M4. server → client: INVITE

When the server receives RESPONSE message from the user, the server computes \(response^* = h(\text{username}||\text{realm}||k)\) and then verifies whether \(response^* = response\). If the equality holds, the server sends the INVITE message and accepts the connection.

After finishing the mutual authentication, both the user and the server compute the common session key \(k = H_3(U^x||R^x_A||R^x_S)\) for their subsequent communication.
2.4 Password change phase

P1. The user can request to change his/her password with the new password $pw_x^*$ after entering a username and a $pw_x$. Next, the user computes a new value of $pw_y^* = h(pw_x^* \oplus N_r)$ to update $pw_y$ into the smart card.

P2. After receiving the demand for password change, the server computes $B_A^* = h(id \oplus pw_y^*)$ and $W_A^* = h(pw_y^*||id) \oplus K_{IDA}$. The new value is stored to the smart card by the server.

3 Cryptanalysis of Yeh et al.’s Protocol

This section demonstrates that Yeh et al.’s protocol not only suffers from stolen smart card attack, but also does not provide perfect forward secrecy.

3.1 Stolen smart card attack

Stolen smart card attack means that an attacker who possessed with smart card performs any operation which the smart card and obtains any secret information [13]. Suppose that an attacker Eve obtained a legal user’s smart card. We know that the smart card has the data $\{B_A, W_A, N_r, h(\cdot), H_1(\cdot), H_2(\cdot), H_3(\cdot)\}$ for the user. Then, the attacker Eve can perform the following stolen smart card attack.

A1. Eve selects a candidate password $pw_x^*$.

A2. Eve computes $pw_y^* = h(pw_x^* \oplus N_r)$.

A3. Eve checks if the following equation holds or not

$$B_A = h(id \oplus pw_y^*)$$

If the check passes, then Eve confirms that the guessed password $pw_x^*$ is the correct one.

A4. If it is not correct, Eve chooses another password $pw_x^{**}$ and repeatedly performs above step (3) until

$$B_A = h(id \oplus pw_y^{**})$$

A5. If Eve correctly obtains the user password $pw_x^*$, Eve can extract the secret key $K_{IDA}$ by computing $W_A = h(pw_y^*||id)$, where $pw_y^* = h(pw_x^* \oplus N_r)$. Because $W_A = h(pw_y^*||id) \oplus K_{IDA} \oplus h(pw_y^*||id) = K_{IDA}$, Eve can easily obtain the secret key $K_{IDA}$.
By using $pw_x^*$ and $K_{IDA}$, the attacker $Eve$ can freely perform the user impersonation attack or the server impersonation attack. Therefore, Yeh et al.’s protocol is vulnerable to the above stolen smart card attack.

### 3.2 Perfect forward secrecy problem

Perfect forward secrecy is one of the security notions addressing the session key exposure issues[7, 8]. Perfect forward secrecy means that if a long-term private key (e.g. user password $pw_x$ or system’s private key $qs$) is compromised, this does not compromise any earlier session keys. Suppose that the system’s private key $qs$ and the user authentication key $K_{IDA}$ are compromised, then the attacker $Eve$ can perform the following attack to obtain the common session key $k = H_3(U^x||R^x_A||R^x_S)$ as follows:

A1. From the intercepted client’s REQUEST message $\{\text{username}, T_1, M_A, R^*_A\}$, the attacker $Eve$ can obtain $U_{IDA} = (U_x, U_y)$ by computing $H_1(id)$ from username, where

$$U_{IDA} = H_1(id) = H_1(\text{username}) \quad (3)$$

A2. By using the compromised system’s private key $qs$ and the obtained $U_{IDA}$, $Eve$ can obtain $R_A = (R^x_A, R^y_A)$ by computing

$$R_A = M_A - qs \times t_1 \times U_{IDA} \quad (4)$$

where $t_1 = H_2(T_1)$ and $M_A$ is included in the client’s REQUEST message.

A3. From the intercepted server’s CHALLENGE $\{\text{realm}, T_2, M_S, M_k\}$, $Eve$ can obtain $R_S = (R^x_S, R^y_S)$ by computing

$$R_S = M_S - t_2 \times K_{IDA} \quad (5)$$

where $K_{IDA}$ is the compromised user authentication key, $t_2 = H_2(T_2)$, and $M_S$ is included in the server’s CHALLENGE message.

A4. By knowing the $U_{IDA} = (U_x, U_y)$, $R_A = (R^x_A, R^y_A)$, and $R_S = (R^x_S, R^y_S)$, $Eve$ can compute any past versions of session keys by computing

$$k = H_3(U^x||R^x_A||R^x_S) \quad (6)$$

Therefore, Yeh et al.’s protocol cannot provide the perfect forward secrecy.
4 Conclusions

This paper pointed out that recently proposed Yeh et al.’s robust smart card secured authentication scheme on SIP using elliptic curve cryptography not only suffers from stolen smart card attack, but also does not provide perfect forward secrecy. Further works will be focused on improving the Yeh et al.’s protocol which can be able to provide strong security.

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

References


Received: August 29, 2015; Published: October 12, 2015