Cryptanalysis on Anonymous Two Factor Mutual Authentication with Key Agreement Scheme For Session Initiation Protocol

Yoonsung Choi
Department of Cyber Security, Howon University, Impi-Myeong, Gunsan-Si, 54058 Jeonrubak-Do, Korea

Abstract: In order to solve the limitations of the password-based authentication scheme, various two-factor authentication schemes for session initiation protocol is studied and regarded as good solution. In 2015, Lu proposed secure two-factor authenticated key agreement which provide the anonymity and key agreement for session initiation protocol using elliptic curve cryptography. However, Reddy point out the vulnerability the Lu scheme and proposed an enhanced anonymous two-factor mutual authentication with key-agreement scheme for session initiation protocol in 2016 but there are various security problem yet. So, this study analyzes all phases of Reddy authentication scheme and then shows that their scheme is still vulnerable on off-line password (and identity) guessing attack a DoS attack, wrong password change phase and session key disclosure attack.

Key words: Cryptanalysis, session initiation, protocol, security analysis, authentication, scheme

INTRODUCTION

SIP (Session Initiation Protocol) is a useful communications protocol for signaling and controlling multimedia communication sessions. Recently, SIP is the most widely used for the current unified communications and voice through internet protocols. SIP can establish, alter and terminate the connection between various communication parties. SIP is made for application layer protocol and it is designed to be independent of the underlying transport layer. And SIP is integrating many elements of the hypertext transfer protocol and the simple mail transfer protocol. Moreover, SIP is text-based protocol and is used for requests from clients and responses from servers over public communication (Johnston, 2009; Yang et al., 2005; Franks et al., 1999).

Rosenberg et al. (2002) proposed a challenge response based authentication scheme for SIP. Various researchers studies more efficient and secure authentication schemes for SIP after (Rosenberg et al., 2002)'s scheme (Irshad et al., 2014; He et al., 2012; Wang and Zhang, 2008; Wu et al., 2009). Lu et al. (2015) introduces an anonymous two-factor elliptic curve cryptography based authenticated key agreement scheme for SIP. Lu et al. (2015) asserted that their scheme is secure various against attacks and provides anonymity. However, Reddy et al. (2016) show that Lu et al. (2015)'s scheme is vulnerable on imperfect mutual authentication and agreement, prone to extraction of sensitive information and prone to key-compromise user impersonation attacks. And Reddy et al. (2016) proposed security enhanced elliptic curve cryptography based scheme which provide anonymous two-factor mutual authentication with key agreement scheme for SIP. They claimed their scheme provide user anonymity, mutual authentication, perfect forward secrecy and is more secure on various attacks than diverse authentication schemes including Lu et al. (2015) scheme. However, this study find out the security vulnerabilities such as off-line password (and identity) guessing attack, a DoS attack, wrong password change phase and session key disclosure attack after analyzing all phases by Reddy et al. (2016) authentication scheme.

Review of Reddy et al. (2016)'s scheme: This study proposes, an improved anonymous two-factor authentication with key-agreement for session initiation protocol using elliptic curve cryptography (Reddy et al., 2016). The notations of the proposed protocol are listed in Table 1.

<table>
<thead>
<tr>
<th>Notations</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>A user</td>
</tr>
<tr>
<td>S</td>
<td>A server</td>
</tr>
<tr>
<td>ID_u</td>
<td>Identity of U</td>
</tr>
<tr>
<td>PW_u</td>
<td>Password of U</td>
</tr>
<tr>
<td>SC</td>
<td>Smartcard of U</td>
</tr>
<tr>
<td>r_u, s</td>
<td>Random numbers chosen by U</td>
</tr>
<tr>
<td>Pr_k</td>
<td>Private key of S</td>
</tr>
<tr>
<td>Pub_s</td>
<td>Public key of S</td>
</tr>
<tr>
<td>r_s, p</td>
<td>Random numbers chosen by S</td>
</tr>
<tr>
<td>P</td>
<td>A point on the elliptic curve</td>
</tr>
<tr>
<td>SK</td>
<td>Session key</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>h(·)</td>
<td>A secure one-way hash function</td>
</tr>
<tr>
<td>σ</td>
<td>An exclusive-or operation</td>
</tr>
</tbody>
</table>
MATERIALS AND METHODS

System initialization phase: Before the protocol is ever executed, this scheme computes and shares the secret as follows.

**Step 1:** S generates a point $P$ on an elliptic curve $E(a, b)$ over $\mathbb{F}_p$.

**Step 2:** S selects $h(\cdot)$ and $\text{Pri}_S \cdot z_0$, calculates $\text{Pub}_S = \text{Pri}_S \cdot P$.

**Step 3:** S stores $\text{Pri}_S$ and publishes $\{E(a, b), P, \text{Pub}_S, h(\cdot)\}$.

User registration phase: For a user $U$, this phase is performed only once when $U$ registers itself with the server. Algorithm 1 illustrates how the phase works and its description follows.

**Algorithm 1 (Reddy et al., 2016)'s user registration phase:**

User $U$

\[
(\text{ID}_0, \text{PW}_U)
\]

Choose $\text{ID}_0, \text{PW}_U, r_0, r$

$\text{PID}_0 = h(\text{ID}_0 || r_0)$

$\text{RPW} = h(\text{PW}_U || r_0) \oplus r$

\[
\begin{align*}
V_1 &= r_0 \oplus h(\text{ID}_0 || \text{PW}_U) \\
N' &= N \oplus r = M \oplus h(\text{PW}_U || r_0) \\
V_2 &= h(\text{PID}_0 || h(\text{PW}_U || r_0))
\end{align*}
\]

and then, $U$ stores them on the received smart card SC. Therefore, SC stores $\{N', V_1, V_2, P, \text{Pub}_S, h(\cdot)\}$.

Mutual authentication with key-agreement phase: In mutual authentication with key-agreement phase, user $U$ and server $S$ can authenticate each other and compute a session-key. Algorithm 2 shows the process of mutual authentication with key-agreement phase as:

**Algorithm 2 (Off-line password (and identity) guessing attack of Reddy et al. (2016)'s scheme):**

An adversary steals U's smart card and obtains all of information. So adversary can get $N', V_1, V_2, P, \text{Pub}_S, h(\cdot)$ using physical monitoring. The adversary knows the formulas of parameters

- $V_1 = h(\text{PID}_0 || \text{PW}_U || r_0)$
- $\text{PID}_0 = h(h(\text{ID}_0 || r_0))$
- $\text{PID}_0 = h(h(\text{ID}_0 || r_0))$
- $V_2 = h(\text{PID}_0 || \text{PW}_U)$
- $V_2 = h(\text{PID}_0 || \text{PW}_U)$
- $[\text{Using PID}_0]$ $V_2 = h(h(\text{ID}_0 || r_0)) h(\text{PW}_U || r_0)$
- $[\text{Using PID}_0]$ $V_2 = h(h(\text{ID}_0 || r_0)) h(\text{PW}_U || r_0)$
- The adversary has $V_1, V_2, h(\cdot)$ and does not know $\text{ID}_0$ and $\text{PW}_U$
- $|D_0|$ and $|D_P|$ denote the number of identities in $D_0$ and passwords in $D_0$
- $D_0$ and $D_P$ are very limited in practice such as $|D_0| < |D_P| < 10^9$
- Thus, the adversary can obtain $\text{ID}_0$ and $\text{PW}_U$ from user's smart card

**Step 1:** U inserts smart card SC and inputs own $\text{ID}_0$ and $\text{PW}_U$. SC computes $r_0, \text{PID}_0$:

\[
\begin{align*}
r_0 &= V_1 \oplus h(\text{ID}_0 || \text{PW}_U) \\
\text{PID}_0 &= h(\text{ID}_0 || r_0)
\end{align*}
\]

and checks the accuracy $V_2 = h(\text{PID}_0 || h(\text{PW}_U || r_0))$. If they are same, then the smart card SC generates a random number $a$ and computes $N_0, M, Y$ as:

\[
\begin{align*}
N_0 &= a \cdot P, N_0 = a \cdot \text{Pub}_S, \\
M &= N_0 \oplus h(\text{PW}_U || r_0) \\
Y &= h(\text{PID}_0 || N_0 || M)
\end{align*}
\]

and then, user $U$ sends the Request (AID$_{00}$, $N_0$, $Y$) to server $S$.

**Step 2:** S computes $N'_0, \text{PID}_0, M$ as:

\[
\begin{align*}
N'_0 &= \text{Pri}_S \cdot N_0, \text{PID}_0 = a \cdot N_0 \oplus \text{Pub}_S \\
M &= h(\text{PID}_0 || \text{ID}_0 || k)
\end{align*}
\]
and then, verifies $Y = h(\text{PID}_0||N_0||M)$. If they are same, server $S$ authenticates $U$, otherwise process aborts. $S$ generates a random number $\beta$ and computes $X, N_5$, $\text{SK}_5$, $\text{auth}_5$.

$$X = M \oplus \beta$$
$$N_5 = \beta \times N_0$$
$$\text{SK}_5 = h(\text{PID}_0 \parallel N_5 \parallel \beta)$$
$$\text{auth}_5 = h(\text{SK}_5 \parallel \text{PID}_0 \parallel M)$$

and then, server $S$ sends challenge (realm, $X$, $\text{auth}_5$) to user $U$.

**Step 3**: Using receiving challenge messages, $S$ computes $\beta, N'_5, \text{SK}_0$, as:

$$\beta = M \oplus X, N'_5 = \beta \times N_0$$
$$\text{SK}_0 = h(\text{PID}_0 \parallel N'_5 \parallel \beta)$$

and then verifies $\text{auth}_5$ messages as:

$$\text{auth}_5 = h(\text{SK}_0 \parallel \text{PID}_0 \parallel M)$$

If $\text{auth}_5$ is same to $h(\text{SK}_0 \parallel \text{PID}_0 || M)$, $U$ authenticates $S$ and further computes $\text{auth}_0$:

$$\text{auth}_0 = h(\text{SK}_0 \parallel \text{PID}_0 \parallel \beta)$$

and then, $U$ sends the RESPONSE(realm, $\text{auth}_0$) to server $S$.

**Step 4**: Server $S$ checks $\text{auth}_0 = h(\text{SK}_0 \parallel \text{PID}_0 || \beta)$. If they are same, $S$ accepts for next communication using computed session key $\text{SK}_0 = \text{SK}_5$.

**Password changing phase**: Reddy et al. (2016)'s protocol allows users to freely update their passwords. The password change phase works as:

**Step 1**: $U$ inserts SC and enters the existing user's ID$_0$ and PW$_0$. SC computes $r_0$.

$$r_0 = V_i \oplus h(\text{ID}_0 \parallel \text{PW}_0)$$

and then, SC verifies the computed value and $V_2$ as:

$$V_2 = h(\text{PID}_0 || h(\text{PW}_0 || r_0))$$

If $V_2 = h(\text{PID}_0 || h(\text{PW}_0 || r_0))$ are same, then $U$ derives $M$ as:

$$M = N' \oplus h(\text{PW}_0 \parallel r_0)$$

**Step 2**: A user $U$ selects a new password PW$_{new}$ and computes $\text{RPW}_{new}$, $V_{i_{new}}$, $N_{new}$, $V_{i_{new}}$ as:

$$\text{RPW}_{new} = h(\text{PW}_{new} || r_i) \oplus r$$

$$V_{i_{new}} = r_i \oplus h(\text{ID}_0 || \text{PW}_{new})$$

$$N_{new} = M \oplus h(\text{PW}_{new} || r_i)$$

$$V_{i_{new}} = h(\text{PID}_0 || h(\text{PW}_{new} || r_i))$$

and then, user $U$ replaces the existing values on the received smartcard. Thus, the smartcard contains $\{N_{new}, V_{i_{new}}, V_{2_{new}}, P, \text{Pub}_0, h(\cdot)\}$.

**RESULTS AND DISCUSSION**

**Cryptanalysis on Reddy et al. (2016)'s scheme**: We analyze Reddy et al. (2016)'s anonymous two-factor mutual authentication with key-agreement scheme for session initiation protocol and determine various security vulnerabilities including off-line password (and identity) guessing attack, a DoS attack, wrong password change phase, session key disclosure attack.

**Off-line password (and identity) guessing attack**: Reddy et al. (2016)'s authentication scheme, an adversary can user's identity and password from user's smart card. It is reason that an attacker can extract the stored information from smart card using Simple Power Analysis (SPA), Differential Power Analysis (DPA). Algorithm 2 shows the process of off-line password (and identity) guessing attack by Reddy et al. (2016)'s scheme (Ma et al., 2014). An adversary gets or steals the user's smart card and then the adversary obtain the all of information $(N', V_i, V_2, P, \text{Pub}_0, h(\cdot))$ from the smart card using physical monitoring such as SPA and DPA which is proved by various studies Kocher and Messerges and the adversary can knows the equation of all parameters such as $V_i$, PID$_0$, and $r_i$:

$$V_2 = h(\text{PID}_0 || h(\text{PW}_0 || r_0))$$

$$\text{PID}_0 = h(\text{ID}_0 || r_0), r_i = V_i \oplus h(\text{ID}_0 || \text{PW}_0)$$

Using this equation and the parameter extracted from smart cards, the adversary can modify $V_2 = h(\text{PID}_0 || h(\text{PW}_0 || r_0))$ as follows:

Using $\text{PID}_0 \rightarrow V_2 = h(\text{ID}_0 || r_0) || h(\text{PW}_0 || r_0)$

Using $r_i \rightarrow V_2 = h(\text{ID}_0 || V_i \oplus h(\text{ID}_0 || \text{PW}_0) || h(\text{PID}_0 || V_i \oplus h(\text{ID}_0 || \text{PW}_0)))$
Fig. 1: A DoS attack by Reddy et al. (2016)'s scheme

The adversary has \( V_i, V_s, h(\cdot) \) from user's smart card and so does not know \( ID_o \) and \( PW_o \) on \( V_i = h( h(ID_o||V_i)||h(ID_o||PW_o) ) \). If the adversary can guess the \( ID_o \) and \( PW_o \) because they are both small size, \( D_{id} \) and \( D_{pw} \) defined the number of identities in \( D_{id} \) and the number of passwords in \( D_{pw} \). If \( T_{hash} \) is the running time for Hash, the running time of the aforementioned attack procedure is \( O(D_{id}|D_{pw}|T_{hash}) \) because both password and identity are human-memorable short strings but not high entropy keys. So, \( D_{id} \) and \( D_{pw} \) are often chosen from two corresponding dictionaries of small size. As \( D_{id} \) and \( D_{pw} \) are very limited in practice, if \( D_{id}|D_{pw}|<10^9 \), the aforementioned attack can be completed in polynomial time. Therefore, the adversary can IDs and PW using off-line password (and identity) guessing attack on Reddy et al. (2016)'s authentication scheme.

DoS attack: Reddy et al. (2016)'s scheme does not use the timestamps so they use random number for preventing the replay attack. However, Reddy et al. (2016)'s scheme has problem on a DoS attack. Figure 1 shows the possibility of a DoS attack by Reddy et al. (2016)'s authentication scheme.

An adversary can obtain and intercept the previous authentication message \( \{ \text{AID}_o, N_o, Y \} \) in public communication. Then, the adversary resend \( \{ \text{AID}_o, N_o, Y \} \) after user's authentication phase ends. However, the server cannot figure out that the message is previous message and cannot checks the legitimacy of incoming message because the server cannot check and know the freshness of message before \( auth_o \) is same to \( h(\text{SK}_o||\text{PID}_o||B) \) (\( auth_o=h(\text{SK}_o||\text{PID}_o||B) \)). So, the server executes various operation such as random number generation operation, hash operation, * operation and exclusive or operation before checking whether \( auth_o \) sent by the adversary and computed \( h(\text{SK}_o||\text{PID}_o||B) \) are same. Therefore, the adversary can execute the DoS attack without difficulty (Choi et al., 2014, 2016; Moon et al., 2016).

Wrong password change phase: When a user want to change own password, the user process the password change phase but password change phase of Reddy et al. (2016)'s scheme has procedural problem (Li et al., 2014; Madhusudhan and Mittal, 2012). To change the password, first the user inputs present \( ID_o \) and \( PW_o \). So, the user's smart card computes \( r_o, M \) and verifies \( V_i \) as:

\[
\begin{align*}
  r_o &= V_i \oplus h( ID_o || PW_o ) \\
  V_i &= h( ID_o || PW_o || r_o ) \\
  M &= N \oplus h( PW_o || r_o )
\end{align*}
\]

Then, the user chooses a new password \( PW'_o \) and then have to compute \( RPW'_{new} \) as:

\[
RPW'_{new} = h( PW'_o || r_o ) \oplus r
\]

But the user cannot compute \( RPW'_{new} \) because the user does not know parameter \( r \). This parameter \( r \) is used to protect the \( h( PW_o || r_o ) \) from the server in registration phase. However, parameter \( r \) does not store in the smart card and cannot compute \( r \) using other parameters. A parameter related \( r \) in smart card is only \( N' \) as:

\[
\begin{align*}
  N' &= N \oplus r \\
  N &= M \oplus RPW = h( ID_o || PW_o || r_o ) \oplus h( PW_o || r_o ) \oplus r \\
  \rightarrow N' &= h( ID_o || PW_o || r_o ) \oplus h( PW_o || r_o ) \oplus r \oplus r \\
  \rightarrow N' &= h( ID_o || PW_o || r_o ) \oplus h( PW_o || r_o )
\end{align*}
\]

As above formulas, \( N' \) does not contain the information about \( r \) because the parameter \( r \) is removed by (exclusive or) operation. Therefore, the user by Reddy et al. (2016)'s scheme cannot change the user's password because user cannot compute parameter \( r \).

Session key disclosure attack: Reddy et al. (2016)'s authenticiation is vulnerable on session key disclosure attack. An adversary can compute the session key including previous session key using smart card. Algorithm 3 shows session key disclosure attack on Reddy et al. (2016)'s scheme (Kumari et al., 2013) as:

Algorithm 3 (Session key disclosure attack by Reddy et al. (2016)'s scheme):
- An adversary obtains AID and X in public communication
- An adversary steals user U's smart card
An adversary can extract all information from smart card using SPA, DPA.

**So attacker can get ID_v and PW_v using off-line guessing attack**

An adversary has AID, X, ID_v, PW_v, N' and V

\[ r' = V \oplus h(ID_v || PW_v) \]

\[ PID_v = h(ID_v || r') \]

\[ N'_v = AID_v \oplus h(ID_v) \]

\[ M = N'_v \oplus h(PW_v || r') \]

\[ \beta = M \times X \]

\[ N'_v = \beta \times N'_v \]

\[ SK_v = h(PID_v || N'_v || \beta) \]

An adversary knows all formula's parameter in \( SK_v = h(PID_v || N'_v || \beta) \)

**So an adversary can compute all of previous session key \( SK_v \)**

Reddy et al. (2016)'s scheme, an adversary can obtain all of AID and X (including previous AID, X) in public communication between user and server. The adversary steals user's smart card, then he can extract all information from smart card using power analysis such as SPA, DPA. And the adversary can compute user's ID_v and PW_v using the stored information. So, the adversary has AID, X, ID_v, PW_v, N' and V_v, so he can compute the session key using them as following procedures.

\[ r'_v = V \oplus h(ID_v || PW_v) \]

\[ PID_v = h(ID_v || r'_v) \quad \text{(using computed } r'_v) \]

\[ N'_v = AID_v \oplus h(ID_v) \quad \text{(using computed PID_v)} \]

\[ M = N'_v \oplus h(PW_v || r'_v) \quad \text{(using computed } r'_v) \]

\[ \beta = M \times X \quad \text{(using computed } M) \]

\[ N'_v = \beta \times N'_v \quad \text{(using computed } \beta) \]

\[ \rightarrow SK_v = h(PID_v || N'_v || \beta) \]

Therefore, the adversary knows and computes all equation's parameter of session key \( SK_v = h(PID_v || N'_v || \beta) \). Moreover, the adversary can calculate all of session key including previous session key.

**CONCLUSION**

Reddy et al. (2016) proposed an anonymous two-factor mutual authenticated key-agreement scheme for session initiation protocol using elliptic curve cryptography overcome the existing security problem but their scheme has security problem. So, this study analyze Reddy et al. (2016)'s scheme and point out that this scheme is vulnerable on off-line password (and identity) guessing attack, a DoS attack, wrong password change phase and session key disclosure attack.

**ACKNOWLEDGEMENT**

This research was supported by financial support of Howon University in 2017.

**REFERENCES**


Franks, J., P. Hallam-Baker, J. Hostetler, S. Lawrence and P. Leach et al., 1999. HTTP authentication: Basic and digest access authentication Internet Engineering Task Force, ©Fremont, California, USA.


