An Energy Efficient Authenticated Key Agreement Protocol for SIP-based Green VoIP Networks

Liping Zhang, Shanyu Tang*, Senior Member, IEEE, Shaohui Zhu

School of Computer Science, China University of Geosciences, Wuhan, 430074, China

*Corresponding author: shanyu.tang@gmail.com, carolyn321@163.com, Tel/Fax: +86 (0)27 6784 8563

Abstract—Voice over Internet Protocol (VoIP) is spreading across the market rapidly due to its characteristics such as low cost, flexibility implementation, and versatility of new applications etc. However, the voice packets transmitted over the Internet are not protected in most VoIP environments, and then the user’s information could be easily compromised by various malicious attacks. So an energy-efficient authenticated key agreement protocol for Session Initial Protocol (SIP) should be provided to ensure the confidentiality and integrity of data communications over VoIP networks. To simplify the authentication process, several protocols adopt a verification table to achieve mutual authentication, but the protocols require the SIP server to maintain a large verification table which not only increases energy consumption but also leads to some security issues. Although several attempts have been made to address the intractable problems, designing an energy-efficient authenticated key agreement protocol for SIP-based green VoIP networks is still a challenging task. In this study, we propose an efficient authentication protocol for SIP by using smartcards based on elliptic curve cryptography. With the proposed protocol, the SIP server needs not to store a password or verification table in its database, and so no energy is required for the maintenance of the verification table. Security analysis demonstrates that the proposed protocol can resist various attacks and provides efficient password updating. Furthermore, the experimental results show that the proposed protocol increases efficiency in comparison with other related protocols.
**Key words**—Green networks; VoIP; Session initiation protocol; Authentication; Key agreement; Security.

1. **Introduction**

Internet and communication technologies boost and diversify the development of Voice over Internet Protocol (VoIP) applications. Compared with traditional Public Switched Telephone Networks (PSTNs), VoIP networks attract great attention since they can provide low cost, flexibility implementation and versatility of new applications. So far, more than five million peoples adopt VoIP services, which are provided by Skype, Gtalk, and iPhone etc. However, with the rapid increase of the registered users on VoIP networks, the energy cost of VoIP networks is also growth with an alarming trend. Furthermore, since the voice data transmitted over the VoIP environments are not protected, the privacy and value information of the users could be compromised easily by inactive or active attacks (Wang and Liu, 2011). Session initiation protocol, developed by Internet Engineering Task Force (IETF), is a text-based application layer control protocol for VoIP setup, modification, and termination among participants (Rosenberg et al., 2002). The architecture of the SIP is generally composed of the proxy server, redirect server, user agent, location server, and register server, as well as main network elements. Compared with other signaling protocols such as H.323, SIP is more lightweight and flexible. However, the authentication of SIP is inherited directly from HTTP Digest authentication (Franks et al., 1999), which is vulnerable to several attacks such as impersonation attacks, off-line password guessing attacks, and server-spoofing attacks etc. In an attempt to ensure the confidentiality and integrity of VoIP communication, an energy-efficient authenticated key agreement protocol for SIP should be sought to achieve mutual authentication and key negotiation between the caller and the callee in a VoIP environment. Although several authenticated key agreement protocols have been proposed in the past years, developing an energy-efficient and secure authentication protocol for SIP is still a challenging task. This is because the green VoIP networks require the authentication protocol to satisfy both the security and the efficiency requirements.
To avoid time-consuming operations, the hash function is considered to be the best candidate to use in the design of security measures. Since hash operations are faster than public key cryptography, hash-based authentication protocols meet low computational requirements of green VoIP by reducing the computational energy cost significantly. However, Kilinc and Yanik (2013) demonstrated that hashed-based authentication protocols had some inherent security weakness, so these protocols were very hard to provide strong security for SIP. Several authentication protocols adopt Public Key Cryptography (PKC) to achieve strong security. In order to simplify the authentication process and reduce time consuming operations, a verification table is required to store in the SIP server’s database for verification purposes in most PKC-based protocols. Although these protocols reduce the computational cost, but also arouse some issues inevitably. Compared with the protocols without using verification tables, these protocols are more vulnerable to guessing attacks, stolen-verifier attacks, and server-spoofing attacks due to the usage of the verification tables. Moreover, with the growth of the registered users the storage overhand will became very high. Furthermore, the maintenance of the large verification tables and the password updating process are all energy consuming operations. Obviously, the verification tables should be avoided when designing an energy-efficient authentication protocol for SIP. However, how to design an authentication protocol for SIP to meet both the secure and energy-efficient requirements remains a challenging work.

In this study, our main objective was to design an energy-efficient authenticated key agreement protocol for SIP without using verification tables. Since no verification table needs to store in the SIP server database, the proposed protocol could not only enhance security but also avoid the energy consumption associated with verification table maintenance. In addition, the complexity analysis demonstrated that the proposed protocol reduced the computational cost in comparison with other related work.

The rest of this paper is organized as follows. The related work is presented in Section 2. In Section 3, the proposed protocol is described in detail. Section 4 discusses the security of the proposed protocol. In Section 5, the performance of the proposed protocol is evaluated. And the paper is concluded in Section 6.
2. Related work

Since the original authentication protocol of SIP is vulnerable to off-line password guessing attacks and server-spoofing attacks (Yang et al., 2005), it could not support integrity and confidentiality protection at an acceptable level for VoIP networks. Moreover, their experiment demonstrated that the computational cost on SIP proxy server was very high in the original authentication protocol (Yanik et al., 2008). Consequently, based on original authentication protocol, several improved protocols for SIP have been proposed to strength the security and promote the performance of VoIP communications.

In order to overcome the security weakness of the original authentication protocol of SIP, Yang et al. (2005) proposed a SIP authentication protocol based on Diffie-Hellman key exchange protocol. In their protocol, a hashed password table was stored at the SIP server side, and the hashed password was used to realize mutual authentication and key agreement. However, Jo et al. (2009) argued that Yang et al.’s protocol was still suffered from the off-line password guessing attack and the usage of expensive exponential computation made their design impractical for SIP. To reduce the computational cost, Durlanik and Sogukpinar (2005) proposed a SIP authentication protocol by using elliptic curve cryptography (ECC). Since ECC could achieve the same level security with a smaller key size, their protocol offered better performance in comparison with Yang et al.’s protocol. Unfortunately, Yoon and Yoo (2009) demonstrated that Durlanik et al.’s protocol was vulnerable to the Denning-Sacco attack. Wu and Wang (2009) also constructed an authentication protocol based on ECC. Since a common secret is shared between the IM services identity module (ISIM) and the authentication center (AC), their protocol achieved efficient mutual authentication. However, the protocol proposed by Wu et al. was suffered from off-line stolen-verifier attacks, Denning-Sacco attacks, and password guessing attacks (Yoon et al., 2010b). Based on Wu et al.’s protocol, an improved authentication protocol was proposed by Yoon et al. (2010b) to eliminate security flaws. But Gokhroo et al. (2011) indicated that the improved protocol could not resist off-line password guessing attacks and replay attacks too. Recently, Arshad and Ikram (2013) also proposed an authentication protocol based on elliptic curve discrete logarithm problem for SIP. However, He et al. (2012) demonstrated that Arshad et al.’s protocol was suffered from off-line
password-guessing attacks and then proposed an improvement protocol to overcome the security weakness.

In order to avoid time-consuming operations, Tsai (2009) adopted a one-way hash function to design an efficient authentication protocol for SIP. Since only one-way hash function and exclusive-or operations were used in their protocol, their protocol reduced computational cost significantly. However, Yoon et al. (2010a) demonstrated that Tsai’s protocol could not resist stolen-verifier attacks, off-line password guessing attacks, Denning-Sacco attacks, and failed to achieve perfect forward secrecy. To address these obstacles, Yoon et al. (2010a) proposed a new protocol. Unfortunately, the proposed protocol was vulnerable to stolen-verifier attacks and off-line password guessing attacks (Xie 2012).

Almost all of the authentication protocols mentioned above require storing a password or verification table at the SIP server side. In these protocols, the SIP server verifies the user’s identity by using the passwords or hashed passwords stored in its database. The main merit of these authentication protocols is simple. As shown in Fig.1, since the user’s passwords are stored in the SIP server’s database, the adversary could launch a stolen-verifier attacks and password guessing attacks to obtain the user’s password. Moreover, a privileged-insider of the SIP server could easily steal the identity and password-verifier table from the SIP server and then use these passwords to impersonate a legal user to access other servers. Consequently, these protocols suffer from the insider attack. Furthermore, the required memories of the verification table increase with the number of the registered users. When there are a lot of registered users in the SIP server, the password or verification tables will became very large. Obviously, the maintenance of the verification table and the password updating process are all energy consuming operations which would limit these protocols’ scalability and applicability. Since the verification table stored at the SIP server not only leads to a risk of various attacks but also decreases the applicability for practical use, it should be avoided in the authentication protocol design for green VoIP networks.
In order to eliminate the password or verification tables, Yeh et al. (2013) adopted the smartcard to construct an authentication protocol based on ECC for SIP. However, their protocol involved the time synchronization problem. Furthermore, the computational cost of the protocol was very high due to 12 times of ECC computation operations were involved. Zhang et al. (2013) also proposed an authenticated key agreement protocol based on ECC to avoid the storage of a verification table at the SIP server side. But Irshad et al. (2014) argued that the protocol was suffered from impersonation attacks and then proposed a single round-trip authentication scheme to overcome security flaws. Unfortunately, their protocol was vulnerable to user impersonation attacks (Arshad and Nikooghadam, 2014). Although Arshad et al.’s improved protocol strength the security, a verification table was required to store in the SIP server’s database. Tu et al. (2014) also proposed a new authentication protocol to overcome the weakness of Zhang et al.’s protocol. But, their protocol could not withstand impersonation attacks and password changing attacks (Farash, 2014; Farash and Attari, 2014).

Although several attempts have been made to avoid the usage of the verification table, but designing a secure and energy-efficient authenticated key agreement protocol without using the verification table for SIP is still a challenging task. In this study, an efficient authentication protocol for SIP is proposed by using smartcard based on ECC. Since no verification table needs to store in the SIP server’s database, the
proposed protocol not only enhances the security but also avoids energy consumption for the maintenance of verification tables. In addition, performance analysis demonstrates that the proposed protocol reduces the computational costs in comparison with other related protocols.

3. Energy-efficient authentication key agreement protocol

In this section, we present our energy-efficient authenticated key agreement protocol of SIP in detail. The proposed protocol comprises four phases: initialization phase, registration phase, authentication phase, and password changing phase. The notations adopted throughout the rest paper are summarized in Table 1. Next, our protocol is described in detail as follows, and it is illustrated in Fig. 2.

Table 1 Notations and Terminology

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_i$</td>
<td>User $i$</td>
</tr>
<tr>
<td>$S$</td>
<td>SIP server</td>
</tr>
<tr>
<td>$ID_i$</td>
<td>Identity of the user $U_i$</td>
</tr>
<tr>
<td>$PW_i$</td>
<td>A low-entropy password of $U_i$</td>
</tr>
<tr>
<td>$s$</td>
<td>A high-entropy secret key of $S$</td>
</tr>
<tr>
<td>$p$</td>
<td>A prime power</td>
</tr>
<tr>
<td>$P$</td>
<td>A generator point with the order $n$ over $E_p(a,b)$</td>
</tr>
<tr>
<td>$F_p$</td>
<td>A prime finite field</td>
</tr>
<tr>
<td>$E_p(a,b)$</td>
<td>An elliptic curve equation</td>
</tr>
<tr>
<td>$r, r_1, r_2, r_3, r_4$</td>
<td>High-entropy random numbers</td>
</tr>
<tr>
<td>$SK$</td>
<td>A shared common session key</td>
</tr>
<tr>
<td>$E_k(.)$</td>
<td>A secure symmetric encryption algorithm with the secret key $k$</td>
</tr>
<tr>
<td>$h(.)$</td>
<td>Secure one-way hash function</td>
</tr>
<tr>
<td>$(Q)_x/(Q)_y$</td>
<td>$x$-coordinate value or $y$-coordinate value of elliptic curve point $Q$</td>
</tr>
</tbody>
</table>
3.1 Initialization phase

In this phase, the SIP server $S$ sets up several security parameters used for authentication and key agreement.

*Step S1*: The SIP server $S$ selects an elliptic curve equation $E_p(a,b): y^2 = x^3 + ax + b(\text{mod } p)$ over a prime finite field $F_p$, where $a,b \in F_p$ and $4a^3 + 27b^2 \neq 0(\text{mod } p)$. Next it chooses a base point $P$ over $E_p(a,b)$.

*Step S2*: The server $S$ chooses a high entropy random integer $s$ as its secret key and computes $P_{pub} = sP$.

And then the server $S$ constructs a secure one-way hash function $h(\cdot): \{0,1\}^* \rightarrow \{0,1\}^k$.

*Step S3*: The server $S$ keeps $s$ secret and publishes ${E_p(a,b), P_{pub}, h(\cdot)}$ as its public parameters.

3.2 Registration phase

When a new user $U_i$ wants to register with the SIP server $S$, it performs the following process with the SIP server $S$ to complete the registration process.

*Step R1*: $U_i \rightarrow S: (ID_i, C_1)$

The user $U_i$ first selects its identity $ID_i$ and its password $PW_i$ freely, and chooses a random high entropy random integer $r$. Next, it selects a one-way hash function $h(\cdot): \{0,1\}^* \rightarrow \{0,1\}^k$ and computes $C_1 = h(PW_i \oplus r)$. Then, the user $U_i$ sends ${ID_i, C_1}$ to the SIP server $S$ over a secure channel.

*Step R2*: $S \rightarrow U_i : \text{Smartcard}(C_3)$

After receiving the information, the SIP server $S$ computes $C_2 = h(ID_i \oplus s)$, and $C_3 = C_1 \oplus C_2 = h(PW_i \oplus r) \oplus h(ID_i \oplus s)$ by using its secret key $s$ and the received message from the user $U_i$. After that, the SIP server $S$ records the secure information $C_3$ in the memory of the smart card and delivers this smart card to $U_i$ through a secure channel.
Step R3: Upon receiving the smart card, \( U_i \) writes \( r \) into the smart card secretly. And then the memory of the smart card contains \((C_3, r)\).

3.3 Authentication phase

During the authentication process, the user \( U_i \) and the SIP server \( S \) perform the following steps to achieve mutual authentication and key negotiation.

**Step A1**: \( U_i \to S : REQUEST(ID_i, C_4, C_6) \)

First, the user \( U_i \) inserts its smartcard into the smartcard reader, and enters its identity \( ID_i \) and its password \( PW_i \). Then the smartcard computes \( C_2 = C_3 \oplus h(PW_i \oplus r) = h(ID_i \oplus s) \) by using the input password \( PW_i \) and the secret information \((C_3, r)\) stored in the smartcard. After that, the smartcard chooses a high entropy random integer \( r_1 \) and calculates \( C_4 = r_1P \) and \( C_5 = r_1C_2 \text{Pub} \). And then it selects a random integer \( r_2 \) and computes \( C_6 = h(C_5) \oplus (h(ID_i \oplus s) \oplus r_2 \| (C_5)_x \| (C_5)_y) \), where \((C_5)_x\) and \((C_5)_y\) are \( x \)-coordinate value and \( y \)-coordinate value of elliptic curve point \( C_5 \), respectively. Finally, the user \( U_i \) relays a request message \( REQUEST(ID_i, C_4, C_6) \) to the SIP server \( S \) over a public channel.
Step A2: $S \rightarrow U_i : CHALLENGE(realm, C_7, Auth, r_4)$

Upon receiving the message REQUEST $(ID_i, C_4, C_6)$, the SIP server $S$ adopts its secret key $s$ and the received message $ID_i$ to compute $C_2 = h(ID_i \oplus s)$. After that it retrieves $(h(ID_i \oplus s) \oplus r_2 \parallel (C_5_s), \parallel (C_3_s),)$ from the received message $C_6$ by computing $(h(ID_i \oplus s) \oplus r_2 \parallel (C_5_s), \parallel (C_3_s),)$ = $h(sC_2C_4) \oplus C_6$ via its secret key $s$, the computed $C_2$, and the received message $(C_4, C_6)$. And then it...
checks whether the following equation holds $\left( C_3 \right)^{\oplus 2} \left( sC_2C_4 \right) \equiv \left( sC_2C_4 \right)$. If it is not equivalent, the authentication process stops; otherwise, the SIP server $S$ calculates $C_2 \oplus h(ID_i \oplus s) \oplus r_2$ to obtain the random integer $r_2$. Next it chooses two random integers $(r_3, r_4)$ and then computes $C_7 = r_3P$ and the session key $SK = h(C_4 \| r_3 \| C_4 \| C_7)$. Next, the SIP server $S$ generates an authentication message $Auth_s = h(h(ID_i \oplus s) \| r_2 \| (SK) \| (C_7) \| (SK) \| (C_7))$. Finally it submits a challenge message $CHALLENGE (realm, C_7, Auths, r_4)$ to the $Ui$.

**Step A3: $U_i \rightarrow S : RESPONSE (realm, Auth_u)$**

After receiving the message $CHALLENGE (realm, C_7, Auths, r_4)$, the smartcard adopts $r_1$ and the received message $C_7$ to compute the session key $SK = h(C_4 \| r_1 \| C_4 \| C_7)$. And then it calculates $h(C_3 \| r_2 \| (SK) \| (C_3) \| (SK) \| (C_3) \| (SK))$ and verifies whether it is equal to the received authentication message $Auths$. If true, the user $U_i$ sets $SK$ as the shared session key and generates the authentication information $Auth_u = h((SK) \| (r_4 + 1) \| (SK))$; otherwise, it terminates the authentication session. Finally, the user $U_i$ sends a response message $RESPONSE (realm, Auth_u)$ to the SIP server $S$.

**Step A4: After receiving the message $RESPONSE (realm, Auth_u)$, the SIP server $S$ checks whether the following equation holds $Auth_u \oplus 2 = h((SK) \| (r_4 + 1) \| (SK))$.** If not, it stops the authentication process; otherwise, it sets $SK = r_3r_4P$ as the shared session key with the user $U_i$.

3.4 Password changing phase

During the password changing phase, the user $U_i$ can change its password $PW$ freely and securely. The steps of the password changing phase are executed as follows and are shown in Fig. 3.

**Step P1: $U_i \rightarrow S (V)$**

If the user $U_i$ wants to change its password, it chooses a new password $PW_i^*$, a new random integer $r^*$ and a nonce $R$ for freshness verification. Next, it inputs its old password and calculates $Z = h(PW_i \oplus r) \oplus C_3$ and $V = E_{(SK)_h}(h(PW_i^* \oplus r^*) \| ID \| R \| Z)$, where $E_{(SK)_h}(\cdot)$ is an encryption function
with the $x$-coordinate of elliptic curve point $SK$ as an encryption key encrypts. Finally, the user $Ui$ submits $V$ to the SIP server $S$.

*Step P2: $S \rightarrow Ui : (W)$*

Upon receiving the message, the SIP server $S$ decrypts the received message $V$ by using its session key $SK$ and calculates $h(ID_i \oplus s)$ by using its secret key $s$ and the decrypted value $ID_i$. And then it verifies whether the following equation holds $h(ID_i \oplus s) = Z$. If not, it refuses the password updating requirement; otherwise, it computes a new secret value $C'_3 = h(PW'_i \oplus r^*) \oplus h(ID_i \oplus s)$ and an encryption value $W = E_{(SK)}(C'_3 || h(C'_3 || (R + 1)))$. And then it submits $W$ to the user $Ui$.

*Step P3: After receiving the message $W$, the user $Ui$ decrypts the message and checks whether the authentication tag $h(C'_3 || (R + 1))$ is valid. If true, it replaces the old values $(C_3, r)$ with $(C'_3, r^*)$; otherwise, it stops the password updating process.*

<table>
<thead>
<tr>
<th>User $Ui$</th>
<th>Smartcard</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Password Changing Phase:</strong></td>
<td><strong>Smartcard</strong></td>
</tr>
<tr>
<td>Select $PW'_i, r^*, R$</td>
<td><strong>Decrypt $V$</strong></td>
</tr>
<tr>
<td>Compute $Z = h(PW'_i \oplus r) \oplus C_3, V$</td>
<td><strong>Encrypt $V$</strong></td>
</tr>
<tr>
<td>$(V)$</td>
<td><strong>Compute $h(ID_i \oplus s)$</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Verify $h(ID_i \oplus s) = Z$</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Compute $C'_3, W$</strong></td>
</tr>
</tbody>
</table>
| **Decrypt $W$** | **Update $(C_3, r)$ with $(C'_3, r^*)$**

![Fig. 3. Password updating phase](image)

### 4. Security analysis

In this section, we evaluate the security of our proposed protocol by analyzing some possible attacks.

1) Replay attacks
Suppose an adversary Bob obtains previous request message \( REQUEST (ID_i, C_4, C_6) \) in \( Step \ A1 \) and replays it to the SIP server \( S \). In the proposed protocol, the SIP server \( S \) will detect this replay attack easily in \( Step \ A3 \) by checking the authentication information \( Auth_u \). This is because Bob cannot correctly guess the session key \( SK \) from the intercepted information to construct a valid \( Auth_u \). To generate a valid session key \( SK \), Bob needs to extract \( r_1 \) from \( C_4 \) or retrieve \( r_3 \) from \( C_7 \), which is equivalent to solving an instance of elliptic curve discrete logarithm problem. For the same reason, Bob cannot obtain \( r_1 \) from the intercepted message \( C_6 \). In addition, since \( SK \) is protected by a secure one-way hash function, Bob cannot get it from \( Auth_u \).

Next, we assume that the previous message \( CHALLENGE (realm, C_7, Auth_u, r_4) \) is intercepted by an adversary Bob and then Bob transforms this message to the user \( U_i \). This attack will be found when the user \( U_i \) verifies the equation \( \sum_{s} r_2 \sum_{(SK)} \sum_{(C_5)} = h(C_2) \) in \( Step \ A3 \). For the same reason, Bob cannot pass this verification process without the knowledge of \( SK \) and the SIP server’s secret key \( s \).

Therefore, Bob cannot launch the replay attack successfully in the proposed protocol.

2) Man-in-the-middle attacks

In the proposed protocol, mutual authentication is provided to resist Man-in-the-middle attacks. If an adversary Bob attempts to impersonate \( U_i \) to establish an independent connection and share a session key with \( S \), he needs to pass the verification process of the SIP server \( S \). To pass this verification process, Bob needs to generate a valid session key \( SK \). When Bob tries to extract \( r_1 \) or \( r_2 \) to construct \( SK \), he faces the elliptic curve discrete logarithm problem. So Bob cannot pass the verification of the SIP server \( S \). For the same reason, Bob cannot go through the verification process of \( U_i \) without the knowledge of \( SK \) and the SIP server’s secret key \( s \). So he cannot impersonate \( S \) to share a session key and make an independent connection with the user \( U_i \).

The above analysis illustrates that the proposed protocol can resist the man-in-middle attack.

3) Modification attacks

Assume that an adversary Bob modifies the \( REQUEST \) message and submits the fraud message \((ID_i, C_4, C_6)\) to the SIP server \( S \). However, this modification can be found easily when SIP server \( S \)
checks the equation \((C_5)_x \parallel (C_5)_y = (sC_2C_4)_x \parallel (sC_2C_4)_y\). This is because Bob cannot generate a valid \(C_5\) without the knowledge of key \(s\). Therefore, Bob cannot launch the modification attack successfully by fabricating the \textit{REQUEST} message.

If Bob modifies a \textit{CHALLENGE} message and sends this forgery \((\text{realm}, C_i', \text{Auth}_i')\) to the user \(U_i\). However, the user \(U_i\) will detect this attack, since Bob cannot construct a valid \(\text{Auth}_i'\) and pass the equation verification of \(\text{Auth}_i = h(C_2 \parallel (SK)_x \parallel (C_5)_x \parallel (SK)_y \parallel (C_5)_y)\) without the knowledge of \(SK\) and the SIP server’s key \(s\). So, Bob cannot impersonate the SIP server by modifying the \textit{CHALLENGE} message.

Suppose Bob modifies the message \textit{RESPONSE} \((\text{realm}, \text{Auth}_u')\) and submits it to the SIP server \(S\). Since Bob cannot guess the session key \(SK\) correctly, this impersonating attack will be found by checking the \(\text{Auth}_u'\) value with the computed value \(h((SK)_x \parallel (r_1 + 1) \parallel (SK)_y)\) in Step A4.

Therefore, the modification attack is invalid in the proposed protocol.

4) Denning-Sacco attacks

Suppose an adversary Bob compromises the previous session key \(SK = h(r_1P \parallel r_2P \parallel r_3P)\) and tries to obtain the user \(U_i\)’s password \(PW_i\) and the SIP \(S\)’s secret key \(s\). Since the session key \(SK\) is constructed by three elliptic curve points and is not connected with the user \(U_i\)’s password \(PW_i\) or the SIP server’s private key \(s\). Bob cannot obtains the secret long-term privacy key \(PW_i\) or \(s\) by compromising an old session key \(SK\). In addition, in each session a fresh session key is generated by using \(r_1P, r_1r_3P, \) and \(r_3P\), where the integer \(r_1\) is chosen by the user \(U_i\) and the integer \(r_3\) is selected by the SIP server \(S\) randomly. Since the session key \(SK\) is not connected with each other, the adversary Bob cannot figure out other session keys with an old session key.

Therefore, the proposed protocol can resist Denning-Sacco attacks.

5) Stolen-verifier attacks
There is no password or verification table needed to be stored in the SIP server side. Consequently, the adversary Bob cannot steal the user’s personal information by launching an attack to obtain the verification table stored in the SIP server database. So, in the authentication process, the adversary cannot impersonate the user $U_i$ to cheat the SIP server $S$ by using the stolen information stored in the SIP server database.

Therefore, the proposed protocol can resist the stolen-verifier attack successfully.

6) Offline dictionary attacks without the smart card

If an adversary Bob intends to perform an offline dictionary attack, and he obtains all the messages during the authentication process. Since the messages transmitted between the user $U_i$ and the SIP server $S$ do not include any information about the user $U_i$’s password $PW_i$, the adversary Bob cannot determine whether each of his guessed passwords is correct or not by using the intercepted information.

Therefore, the proposed protocol can resist the offline dictionary attack without the smart card.

7) Offline dictionary attacks with the smart card

Suppose, an adversary Bob compromises the user’s secret information $(C_3, r)$ stored in the smart card and records all the messages transmitted during the authentication process. In this case, Bob possess additional information $(C_3, r)$ stored in the smartcard. However, Bob cannot obtain $h(PW_i \oplus r)$ without the knowledge of the SIP server $S$’s secret key $s$. So, the extra information $(C_3, r)$ cannot help Bob to guess the user $U_i$’s password correctly.

Therefore, the offline dictionary attack with the smart card cannot be launched successfully in the proposed protocol.

8) Insider attacks

Since no password or verification tables are needed to be stored at the SIP server side in the proposed protocol, a privileged-insider of the SIP server cannot access other servers successfully by stealing the identity and password-verifier table from the SIP server $S$.

Therefore, the proposed authentication process can resist insider attacks successfully.

9) Password disclosure attacks
In our protocol, the user $U_i$ submits $C_1 = h(PW_i \oplus r)$ instead of its original password $PW_i$ to the SIP server $S$. Since the real password $PW_i$ is protected by a high entropy random integer $r$, the SIP server $S$ cannot obtain the user $U$’s real password in the registration phase.

Therefore, the proposed protocol can resist the password disclosure attack.

10) Session key security

In the proposed protocol, only the user $U_i$ and the SIP server $S$ know the session key at the end of the key negotiation process $SK = h(r_i P \| r_j P \| r_3 P)$. This is because Bob cannot correctly guess $r_1 r_3 P$ from the intercepted information to construct a valid $SK$. To generate a valid session key $SK$, Bob needs to extract $r_1$ from $C_4$ or retrieve $r_3$ from $C_7$, which is equivalent to solving an instance of elliptic curve discrete logarithm problem. So, the session key $SK$ is not known by anyone but only the user $U_i$ and the SIP server $S$.

Therefore, session key security is provided in the proposed protocol.

11) Known-key security

In the proposed protocol, the user $U_i$ and the SIP server $S$ choose two random integers $r_1$ and $r_2$ respectively in each session process. Since the two integers are different in every session key negotiation process, the $SK$ of each session is not connected with other session keys. Since a different session key is generated in each session, an adversary Bob cannot figure out another session key $SK' = h(r'_i P \| r'_j P \| r'_3 P)$ by using a compromised session key $SK = h(r_i P \| r_j P \| r_3 P)$. So, a unique session key $SK$ is generated between the user $U_i$ and the SIP server $S$ in each run of the authentication process.

Therefore, the proposed protocol provides known-key security successfully.

12) Perfect forward secrecy

Assume that an adversary Bob compromises the user $U_i$’s password $PW_i$ and the SIP server $S$’s secret key $s$. And then it attempts to find the previous session key $SK = h(r_i P \| r_j P \| r_3 P)$. However, without the knowledge of $r_1$ or $r_2$, he cannot construct the previous session key $SK$. This is because the two integers are protected by elliptic curve discrete logarithm problem. In addition, Bob cannot extract $SK$ directly from $Auth_s$ or $Auth_u$ since it is protected by a one-way hash function. So, even if the user $U_i$’s password
and the SIP server $S$’s secret key $s$ are compromised by the adversary Bob, the previous session keys would not be compromised. Therefore, the proposed protocol can provide perfect forward secrecy.

13) Mutual authentication

In the proposed protocol, the user $U_i$ and the SIP server $S$ can verify the identity of each other via $Auth_s$ and $Auth_u$. Therefore, the proposed protocol can provide mutual authentication.

14) Security chosen and update password

In the proposed protocol, the user can freely choose her or his password in the registration phase. In addition, a password updating function is provided for users to change their passwords easily and freely. Furthermore, even if the smart card was stolen or lost, other person could not change the password without knowing the user’s password.

5 Performance comparisons

In our study, we compared the proposed protocol with other related protocols in terms of functionality and computational cost. Since no password or verification table is stored in the SIP server’s database, the proposed protocol avoids energy consumption for maintenance of the verification table. Furthermore, several attacks associated with the verification table could be resisted successfully with the proposed protocol. As shown in Table 2, the protocols proposed by Tsai (2009), Arshad and Ikram (2013), and He et al. (2012) all required the SIP server storing a verification table and did not provide efficient password updating. In addition, Arshad’s protocol was suffered from offline password guessing attacks. And Tsai’s protocol could not resist offline password guessing attacks, Denning Sacco attacks, and stolen verifier attack, so it was weaker than other related protocols. The protocol proposed by Tu et al. (2014) did not need to store a verification table in the SIP server database, but it was vulnerable to modification attacks and could not provide password updating. Although Yeh et al.’s protocol satisfied most of the security requirements, it involved the time synchronization problem. As shown in Table 2, the proposed protocol could not only secure against several attacks but also provide some unique features such as no password or verification table needed, no time synchronization issue, and efficient password updating, etc.
Table 2. The functionality comparisons between the proposed protocol and others

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Replay attack resist</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Modification attack resist</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Offline password guessing attack resist</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Stolen verifier attack resist</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Denning Sacco attack resist</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Mutual Authentication</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Password updating</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>No verifier table</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>No time synchronization</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

We also compared the computational cost of the proposed protocol with other related protocols. To simulate a practical environment, the SIP server and the client were installed on two PCs over the local area network in our experiments. The hardware platform for client was Intel Pentium G630 processor with 4 GB memory which offered maximum clock speeds of 2.7 GHz. The Intel G850 processor was adopted at the SIP server side, which offered maximum clock speeds of 2.90 GHz and 4 GB memory. Furthermore, a NIST/SECG-standard elliptic curve over a 521 bits prime field and SHA-1 as a one-way hash function were adopted in our experiments. Then some notations are defined as follows:

1. $T_m$: the time for executing a scalar multiplication operation of elliptic curve;
2. $T_a$: the time for executing a point addition operation of elliptic curve;
3. $T_h$: the time for executing a one-way hash function (string to number);
4. $T_{H}$: the time for executing a one-way hash function (string to point);
5. $T_v$: the time for executing a modular inversion operation.

As shown in Table 3, in the registration phase, two hash operations are needed to compute $C_1$ on the user side and $C_2$ on the SIP server side. Since only two hash operations are adopted in the registration phase, the execution time of this process is estimated to be 0.012 ms.
In the authentication phase, the user side needs three scalar multiplication operations of elliptic curve to obtain $C_4, C_5$, and $r_1C_7$; four hash operations to generate $h(C_5), SK, Auth_s$ and $Auth_u$. The SIP server side requires three scalar multiplication operations of elliptic curve to compute $sC_2C_4, C_7$ and $r_3C_4$; and five hash operations to obtain $C_2, h(sC_2C_4), SK, Auth_s$ and $Auth_u$. The experimental results showed that 69.12 ms was required during the authentication process in the proposed protocol.

According to Yeh et al.’s protocol, the user side requires one hash operation to compute $h(pw_x \oplus N_r)$ during the registration process, and the SIP server side needs two hash operations to obtain $h(id \oplus pw_x)$ and $h(id \parallel pw_x)$: a scalar multiplication operation of elliptic curve and a hash operation to compute $q_x \times H_i(id)$. In Tu et al.’s protocol, the user side needs one hash operation to obtain $h(pw_i | a)$ during the registration process, and the SIP server side requires one scalar multiplication operation of elliptic curve and one hash operation to compute $(h(pw_i | a) + h(username | s))P$.

### Table 3. Computational comparisons between our protocol and others

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Registration</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User side</td>
<td>$T_h$</td>
<td>$T_h$</td>
<td>$T_h$</td>
<td>$T_h$</td>
<td>$T_h$</td>
<td></td>
</tr>
<tr>
<td>Server side</td>
<td>$2T_h$</td>
<td>$2T_h$</td>
<td>$2T_h + T_H + T_m$</td>
<td>$T_m + T_h$</td>
<td>$T_h$</td>
<td></td>
</tr>
<tr>
<td>Execute time</td>
<td>0.012ms</td>
<td>0.012ms</td>
<td>10.212ms</td>
<td>9.860ms</td>
<td>0.012ms</td>
<td></td>
</tr>
<tr>
<td><strong>Authentication</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User side</td>
<td>$4T_H$</td>
<td>$2T_m + 3T_h$</td>
<td>$3T_m + 3T_h$</td>
<td>$4T_m + 2T_m + 6T_h$</td>
<td>$3T_m + T_s + 4T_h$</td>
<td>$3T_m + 4T_h$</td>
</tr>
<tr>
<td>Server side</td>
<td>$3T_H$</td>
<td>$3T_m + T_s + 3T_h$</td>
<td>$3T_m + 3T_h$</td>
<td>$3T_m + 2T_m + 5T_h$</td>
<td>$3T_m + 4T_h$</td>
<td>$3T_m + 5T_h$</td>
</tr>
<tr>
<td>Execute time</td>
<td>0.724 ms</td>
<td>57.612 ms</td>
<td>69.084 ms</td>
<td>98.620 ms</td>
<td>71.096 ms</td>
<td>69.12 ms</td>
</tr>
</tbody>
</table>

As shown in Fig. 4, compared with other protocols, Tsai’s protocol achieves the best performance, since only one-way hash function and exclusive-or operations are used during the authentication process.
Although Tsai’s protocol reduces the computational cost significantly, their protocol has some security weaknesses. So their protocol is not suitable for VoIP networks. The experimental results show that our protocol is as efficient as Arshad and Ikram’s protocol and He et al.’s protocol, which needs to store a verification table in the SIP server’s database. Since our protocol avoids the energy consumption from the maintenance of the verification table, it is more suitable for green VoIP networks in comparison with Arshad and Ikram’s protocol and He et al.’s protocol. Moreover, compared with Yeh et al.’s protocol and Tu et al.’s protocol, our proposed protocol possesses better performance by reducing the scalar multiplication operations of elliptic curve and by eliminating the point addition operations of elliptic curve.

![Fig. 4 Execution time comparisons between our protocol and others](image)

6 Conclusion

In this study, we proposed an energy-efficient authentication protocol for SIP without using a verification table. Based on ECC, the proposed protocol realized mutual authentication and key negotiation by using password and smartcard. Since no password or verification table was required to store at the SIP server side, the proposed protocol avoided the energy consumption for the maintenance of a large verification table. Furthermore, the proposed protocol could resist several attacks associated with verification tables, such as insider attacks, stolen verifier attacks, and password guessing attacks. Security analysis demonstrated that our protocol was more secure than the related protocols. And the experimental results showed that the proposed protocol reduced the computational cost in comparison with the
protocols without using verification tables. Therefore, the proposed authentication protocol is more suitable for green VoIP networks.

Acknowledgment

This work was supported by the National Natural Science Foundation of China [Grant numbers 61303237, 61272469], the Wuhan Scientific Research Program [Grant number 2013010501010144]; China Postdoctoral Fund [grant number 2012194091], and the Fundamental Research Funds for the Central Universities [Grant number 2013199037].

References


