Computer Communications 33 (2010) 1674–1681

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

A secure and efficient SIP authentication scheme for converged VoIP networks

Eun-Jun Yoon [a], Kee-Young Yoo [a], Cheonshik Kim [b], You-Sik Hong [c], Minho Jo [d,*], Hsiao-Hwa Chen [e,**]

[a] School of Electrical Engineering and Computer Science, Kyungpook National University, 1370 Sankyuk-Dong, Buk-Gu, Daegu 702-701, South Korea
[b] Department of Computer Engineering, College of Electronics and Information Engineering, Sejong University, 98 Gunja-Dong, Gwangjin-Gu, Seoul, 143-747, South Korea
[c] Department of Computer Science, SangJi University, 220-702, 660 Usandong, Wonjusi Gangwondo, South Korea
[d] School of Electrical Engineering, Korea University, 5-ka, Anam-dong, Seongbuk-Gu, Seoul 136-701, South Korea
[e] Department of Engineering Science, National Cheng Kung University, 1 Da-Hsueh Road, Tainan City 70101, Taiwan, ROC

ARTICLE INFO

Article history:
Available online 30 March 2010

Keywords:
SIP
Converged VoIP network
Ubiquitous computing
Cryptography
Cryptanalysis

ABSTRACT

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). SIP is a powerful signaling protocol that controls communications on the Internet for establishing, maintaining and terminating sessions. The services that are enabled by SIP are equally applicable to mobile and ubiquitous computing. This paper demonstrates that recently proposed SIP authentication schemes are insecure against attacks such as off-line password guessing attacks, Denning-Sacco attacks and stolen-verifier attacks. In order to overcome such security problems, a new secure and efficient SIP authentication scheme in a converged VoIP network based on elliptic curve cryptography (ECC) is proposed and it works to exploit the key block size, speed, and security jointly.

1. Introduction

In 2002, Rosenberg et al. [1] in the Internet Engineering Task Force (IETF) Multi-Party Multimedia Session Control (MMUSIC) Working Group proposed Session Initiation Protocol (SIP) as the IP-based telephony protocol [1–8]. SIP, as described in RFC 3261 [1], has been chosen by the Third-Generation Partnership Project (3GPP) as the protocol for multimedia applications in 3G mobile networks [9,10]. With the widespread application of the Voice over IP (VoIP) in the Internet, SIP is receiving a lot of attention and the security of SIP is thus becoming increasingly important.

Unlike many legacy Time Division Multiplex (TDM) voice networks that are physically separated from data-centric networks, the new VoIP networks allow the convergence of networks, and the SIP-based next generation networks possess the advantages of IP voice, web enabled control, open/standard-based features and converged network capabilities. Because SIP is a text-based signaling protocol, it can widely be used for controlling multimedia communication sessions such as voice and video calls over Internet protocols such as Hyper Text Transport Protocol (HTTP) and Simple Mail Transport Protocol (SMTP) [11]. Therefore, SIP is more lightweight and flexible than the other signaling protocols (such as H.323). Like HTTP and SMTP, SIP is a request-response protocol, meaning that it makes a request of a server and then awaits a response. Therefore, to develop a secure user authentication scheme is a critical issue in SIP-based services. For example, suppose that a client *A* (i.e., caller) wants to establish a SIP voice call to the server *B* (i.e., callee). In this case, *A* must verify that he/she is connected exactly to SIP user agent of *B* not to an attacker. 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.

To date, various SIP authentication schemes have been proposed to strengthen the security. In 2005, Yang et al. [12] pointed out that the procedure of the original SIP authentication scheme based on HTTP digest authentication is vulnerable to the off-line password guessing attacks and the server spoofing attacks. They proposed a secure SIP authentication scheme based on the Diffie–Hellman key exchange algorithm [13], which works based on the difficulty of Discrete Logarithm Problem (DLP) to resist attacks. However, Yang et al.'s scheme is not suitable for devices with a low computational power because of the high computational costs. Based on Yang et al.'s scheme, Durlanik et al. [14] proposed an efficient SIP authentication scheme by using elliptic curve cryptosystem (ECC) [15–18]. ECC presents an attractive alternative cryptosystem, because its security is

* Corresponding author. Tel.: +82 2 3290 4764; fax: +82 2 928 9109.
** Corresponding author. Tel.: +886 6 2757575x63320; fax: +886 6 2766549.
E-mail addresses: minhojo@korea.ac.kr, minhojo@gmail.com (M. Jo), hshwchen@ieee.org, hshwchen@mail.ncku.edu.tw (H.-H. Chen).

0140-3664/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2010.03.026

based on the elliptic curve discrete logarithm problem (ECDLP) and it operates over a group of points on an elliptic curve. It can offer a level of security comparable to the classical cryptosystems that use much larger key sizes. Therefore, Durlanik et al.'s scheme can reduce the total execution time and memory requirements in comparison with Yang et al.'s scheme. In 2009, Wu et al. [19] also proposed an SIP authentication scheme based on elliptic curve cryptography (ECC), which overcomes the inherent weaknesses of the existing SIP authentication schemes to achieve authentication and a shared secrecy at the same time and provide provable security in the Canetti–Krawczyk (CK) security model [20]. They claimed that their scheme not only has the security attributes required by the SIP standard with minimal changes, but also is designed to provide data confidentiality, data integrity, authentication, access control, and perfect forward secrecy jointly. It is secure against well-known cryptographical attacks such as replay attacks, off-line password guessing attacks, man-in-the-middle attacks, and server spoofing attacks. Compared with the previous schemes [12,14], Wu et al.'s scheme is more efficient and preferable for applications which require low memory and rapid transactions.

Nevertheless, both Durlanik et al.'s and Wu et al.'s SIP authentication schemes are still vulnerable to off-line password guessing attacks, Denning-Sacco attacks, and stolen-verifier attacks [21–23]. This paper demonstrates the vulnerability of the two schemes to these attacks, and then proposes a secure and efficient SIP authentication scheme in a converged VoIP network based on ECC in order to overcome those security problems and exploit the key block size, speed, and security. As a result, the proposed SIP authentication scheme can resist those attacks, while also providing high security and efficiency. It can be executed faster than previously proposed schemes.

The remainder of this paper is outlined as follows. Section 2 introduces the SIP architecture and authentication procedure. Sections 3 and 4 briefly review Durlanik et al.'s and Wu et al.'s SIP authentication schemes and then show their security problems, respectively. The proposed scheme is presented in Section 5, while Sections 6 and 7 discuss the security and efficiency of the proposed scheme, respectively, followed by the conclusions presented in Section 8.

2. Sip architecture and authentication procedure

This section introduces the SIP architecture and SIP authentication procedures [12,14].

2.1. SIP architecture

The SIP architecture has the client–server features of HTTP based on uniform resource allocators and uniform resource identifiers. A text-encoding scheme and header format were proposed as those in SMTP. That is, SIP reuses SMTP headers such as To, From, Date, and Subject [14]. A Universal Resource Identifiers (URIs) is used for identifying users. Therefore, a URI identifies resources on the Internet. A Uniform Resource Locator (URL) is used to identify Web sites. The URI used by SIP incorporates a phone number or name (e.g., SIP: user1@knu.ac.kr) which can make it easier to read the SIP addresses.

The SIP architecture is mainly composed of a user agent client, proxy server, redirect server, register server, and location server [12]. The function of each component is described as follows.

- *User agent*: A user agent is a logical entity such as a caller (the user agent client (UAC)) or a callee (the user agent server (UAS)).
- *Proxy server*: A proxy server forwards a request and response between a caller and callee. When the proxy server receives a request, it forwards the request to the current location of the callee and then forwards the response from the callee to the caller.
- *Redirect server*: When a redirect server receives a request, it informs the caller about the current location of the callee. Then the caller contacts the callee directly.
- *Register server*: When a user agent changes its location, the user agent sends a register request to the register server to update its current location. In brief, the register server helps the user agent update the information of the user agent's location in the location server.
- *Location server*: The responsibility of the location server is to maintain information on the current location of the user agent. It also services the proxy server, redirect server, and register server for them to look up or register the location of the user agent.

A basic SIP registration and SIP session are presented in Figs. 1 and 2, respectively. Here, INVITE (Initiate a session), REGISTER (Register a user's location), BYE (Terminate a session), ACK (Acknowledge session initiation), CANCEL (Cancel a pending request), and OPTIONS (Query server capabilities) messages are the six original SIP messages in the protocol stack [14].

2.2. SIP authentication procedure

SIP authentication security is based on the challenge–response mechanism [12]. Before the authentication procedure starts, the

Fig. 1. SIP registration procedure.

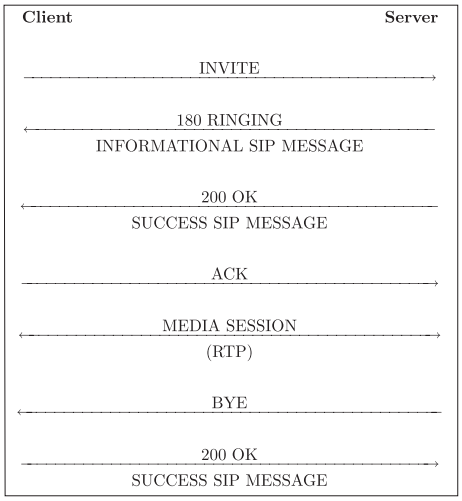

Fig. 2. SIP session procedure.

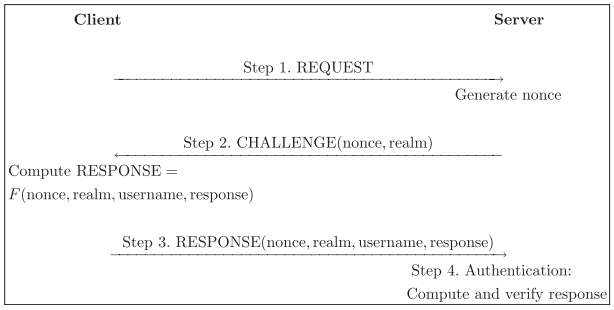

Fig. 3. SIP authentication procedure.

client pre-shares a password with the server. The pre-shared password is used to verify the identity of the client or the server. Fig. 3 shows an example procedure of the authentication mechanism in SIP. The protocol proceeds as follows.

(1) Client → Server: REQUEST
The client sends a REQUEST to the server.
(2) Server → Client: CHALLENGE (nonce, realm)
The server generates a CHALLENGE that includes a nonce and the client's realm. It is noted that the realm is used to prompt the username and password. Then the server sends a CHALLENGE back.
(3) Client → Server: RESPONSE (nonce, realm, username, response)
The client computes a response = F(nonce, realm, username, response). Note that $F(\cdot)$ is a one-way hash function and is used to generate a digest authentication message. Then the client sends the RESPONSE to the server.
(4) According to the username, the server extracts the client's password. Then the server verifies whether or not the nonce is correct. If it is correct, the server computes h(nonce, realm, username, response) and uses it to make a comparison with the response. If they match, the server authenticates the identity of the client.

3. Cryptanalysis of SIP authentication scheme by Durlanik et al.

This section reviews Durlanik et al.'s SIP authentication scheme [14] and then shows that the scheme is vulnerable to the Denning-Sacco attacks and stolen-verifier attacks [21,22]. Notations used in this paper are defined in Table 1.

Table 1
Notations and their explanations.

U	The user agent client (UAC)
S	The user agent server (UAS)
D	A uniformly distributed dictionary of size $\|D\|$
pw	A low-entropy password of U which is randomly chosen from D
x	A high-entropy secret key of S
n	A random nonce generated by U
sk	A shared common session key between U and S
$X \rightarrow Y : M$	X sends a message M to Y
E	Elliptic curve over a finite field $GF(q)$
$E(GF_q)$	An additive group of points on E over a finite field $GF(q)$
P	The generating element (point) of $E(GF_q)$ under consideration
c	A secret random integer chosen by A
os	A secret random integer chosen by S
$F(\cdot)$	A secure one-way hash function, where $h : \{0,1\} \rightarrow \{0,1\}^k$, e.g., $h(x)$ is a secure hash function at the x-coordinate of point $X \in E(GF_q)$
$\oplus$	A bit-wise exclusive-or (XOR) operation

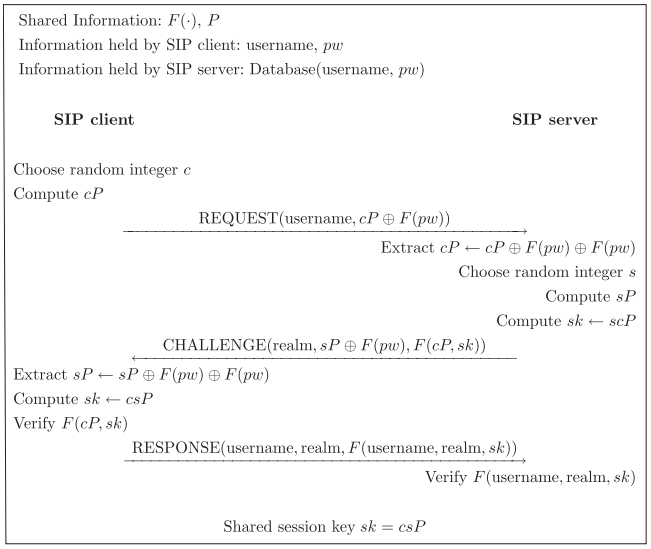

Fig. 4. Durlanik et al.'s scheme.

3.1. Durlanik et al.'s scheme

Fig. 4 illustrates Durlanik et al.'s SIP authentication scheme. If a legal SIP client U wants to login the SIP server S, he/she first inputs his/her username and password pw into the SIP client system. Then, the authentication scheme between U and S proceeds as follows.

(1) $U \rightarrow S$: REQUEST (username, $cP \oplus F(pw)$)
U generates a random integer c, computes $cP \oplus F(pw)$, and then sends it with a request message as *REQUEST* (*username*, $cP \oplus F(pw)$) to S.
(2) $S \rightarrow U$: CHALLENGE (realm, $sP \oplus F(pw), F(cP, sk)$)
Upon receiving the request message, S derives cP by computing $cP \oplus F(pw) \oplus F(pw)$. Then, S generates a random integer s and computes a secret session key $sk = scP$ and a message authentication code $F(cP, sk)$. Finally, S sends a challenge message CHALLENGE (realm, $sP \oplus F(pw), F(cP, sk)$) to U.
(3) $U \rightarrow S$: RESPONSE (username, *realm*, F(*username*, *realm*, *sk*))
Upon receiving the challenge message, U derives sP by computing $sP \oplus F(pw) \oplus F(pw)$ and computes a secret session key $sk = scP$. Then, U computes $F(cP, sk)$ and verifies whether it is equal to the received challenge $F(cP, sk)$. If not, U rejects the server challenge message. Otherwise, U authenticates SIP server S and computes a message authentication code F(username, realm, sk). Finally, U sends a response message RESPONSE (username,realm, F(username, realm, sk) to S.
(4) Upon receiving the response message, S computes F(username, realm, sk) and then verifies whether it is equal to the received response F(username, realm, sk). If not, S rejects the user response message. Otherwise, S authenticates U and accepts the user's login request.

After mutual authentication between U and S, $sk = csP$ is used as a shared session key.

3.2. Denning-Sacco attacks

Dennig-Sacco attack works when an SIP client/server compromises an old shared session key sk and an attacker tries to find a long-term private key (e.g., user password pw) or other session keys. This attack arises from the fact that compromising a fresh

session key sk enables the scheme to be compromised. Such attacks have been known for some time. Refer to the Denning-Sacco attack in [21–23]. In Durlanik et al.'s SIP authentication scheme, the following Denning-Sacco attack is possible.

(1) An attacker records one run of Durlanik et al.'s SIP authentication scheme and somehow obtains the old shared session key $sk = csPG$ between the SIP client and the server.
(2) Since $cP \oplus F(pw)$ and $F(cP, sk)$ are open values from Steps (1) and (2), the attacker can find the long-term private password pw included in $cP \oplus F(pw)$ by performing the following off-line password guessing attack.
   (a) The attacker guesses the secret password pw^* from the password dictionary D.
   (b) The attacker checks if $F(cP, sk) \stackrel{?}{=} F(cP \oplus F(pw) \oplus F(pw^*), sk)$.
   (c) If yes, the attacker has guessed the correct secret password $pw^* = pw$.
   (d) If not, the attacker repeats the above verification process until $F(cP, sk) \stackrel{?}{=} F(cP \oplus F(pw) \oplus F(pw^*), sk)$.
(3) Compromising the user's secret password pw will enable the attacker to impersonate the SIP client/server freely.

The algorithm of the Denning-Sacco attack is given as follows.

Denning-Sacco Attack $(sk, cP \oplus F(pw), F(cP, sk), D)$
{
  for $i := 0$ to $|D|$
  {
    $pw^* \leftarrow D$;
    $cP^* = cP \oplus F(pw) \oplus F(pw^*)$;
    if $F(cP, sk) = F(cP^*, sk)$ **then return** pw^*
  }
}

For example, suppose that an attacker chooses a random nonce e and sends an illegal request message REQUEST (username, $eP \oplus F(pw^*)$) to the SIP server in Step (1) of Durlanik et al.'s SIP authentication scheme. Then, the SIP server will send a challenge message CHALLENGE (realm, $sP \oplus F(pw)$, $F(eP, sk)$) to the attacker. After receiving the challenge message from the SIP server, the attacker can send a response message RESPONSE (username, realm, F(username, $realm$, sk)) to the SIP server by using the compromised user's secret password pw^*. Then, the SIP server will authenticate the attacker by performing the authentication scheme. Therefore, Durlanik et al.'s SIP authentication scheme is obviously insecure against the Denning-Sacco attacks.

3.3. Stolen-verifier attacks

In most existing password authentication schemes, the server stores the user's verifier (e.g., plantext passwords or hashed passwords) rather than the user's bare password in order to reduce the security breach once the server is compromised. Therefore, servers are always the targets of attackers, because numerous customers' secrets are stored in their databases. Stolen-verifier attack [23] means that an attacker who steals a password-verifier from the server can use it directly to impersonate a legitimate user in a user authentication execution. It is noted that the main purpose of an authentication scheme protecting against the stolen-verifier attack is to reduce the immediate danger to the authenticate user. In fact, an attacker who has a password-verifier may further mount a guessing attack.

In Durlanik et al.'s SIP authentication scheme, the password pw of the user, which is stored in the SIP server, can be eavesdropped and then used by the attacker to masquerade as the original user. The authors of Durlanik et al.'s SIP authentication scheme did not explain the stolen-verifier attacks with regard to obtaining the secret data pw, which is stored in a SIP server. This information can allow an illegitimate user to login the SIP server as a legitimate user. Suppose that an attacker has stolen the password pw in the SIP server. Then, he/she can easily impersonate a legal user or the SIP server by performing the authentication scheme.

In the modern life which the Internet has strong influence to people, passwords are the most common means of user authentication on the Internet. For practical applications, password-based authentication schemes are required when making use of the Internet services such as e-learning, on-line polls, on-line ticket-order systems, roll call systems, on-line games, etc. In real applications, users tend to use the same password as above to access several application servers for their convenience. Thus, the attacker may try to use the password pw to impersonate a user to login to other systems that the user has registered with outside this SIP server. If the targeted outside server adopts the normal authentication protocol, it is possible that the attacker can successfully impersonate the user to login to it by using the stolen password pw. Therefore, Durlanik et al.'s SIP authentication scheme is insecure against stolen-verifier attacks.

4. Cryptanalysis of SIP authentication scheme by Wu et al.

This section reviews Wu et al.'s SIP authentication scheme [19] and then shows that the scheme is vulnerable to off-line password guessing attacks [21,22].

4.1. Wu et al.'s scheme

Fig. 5 illustrates Wu et al.'s SIP authentication scheme. Assume that a shared secret password pw and an access point P are established beforehand between the client and server, and pw is stored in an IP Multimedia Services Identity Module (ISIM) which resides on a smart card like a tamper resistant device. There are five steps in the Wu et al.'s SIP authentication scheme listed as follows.

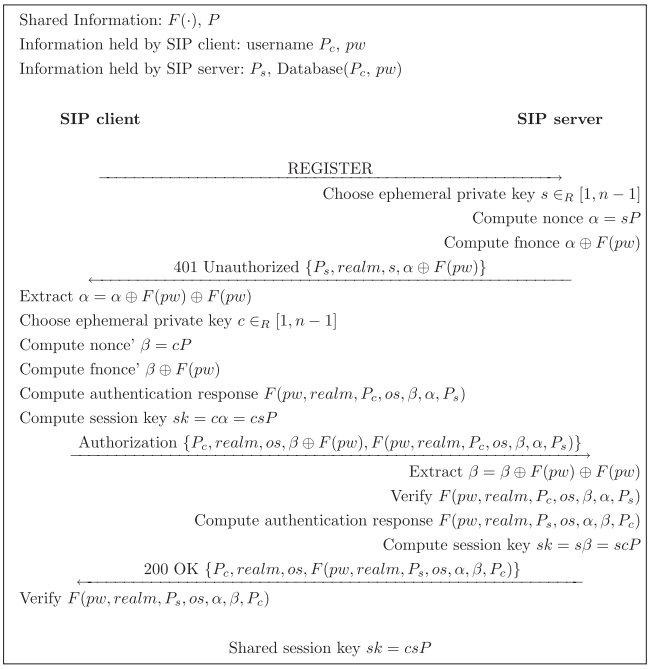

Fig. 5. Wu et al.'s scheme.

(1) $U \rightarrow S$: REGISTER
U makes a request of an SIP service requiring authentication (REGISTER).
(2) $S \rightarrow U$: 401 Unauthorized $\{P_s,\ realm,\ os,\ \alpha \oplus F(pw)\}$
S chooses an ephemeral private key $s \in_R [1, n-1]$, and calculates nonce $\alpha = sP$ and fnonce $\alpha \oplus F(pw)$. Then, it prepares 401 Unauthorized Authentication Required as appropriate. This response is a challenge consisting of an identity of the Server P_s, realm string *realm*, opaque string *os*, and fnonce $\alpha \oplus F(pw)$.
(3) $U \rightarrow S$: Authorization $\{P_c,\ realm,\ os,\ \beta \oplus F(pw),\ F(pw, realm,\ P_c,\ os,\ \beta,\ \alpha,\ P_s)\}$
U chooses an ephemeral private key $c \in_R [1, n-1]$, calculates nonce' $\beta = cP$ and fnonce' $\beta \oplus F(pw)$, and produces an authentication response $F(pw, realm, P_c, os, \beta, \alpha, P_s)$. Then, U derives session key $sk = c\alpha = csP$. Finally, U sends the response along with the username P_c, realm string *realm*, opaque string *os*, and fnonce' $\beta \oplus F(pw)$ in plain text to the entity requesting authentication.
(4) $S \rightarrow U$: 200 OK $\{P_c,\ realm,\ os,\ F(pw, realm, P_s, os, \alpha, \beta, P_c)\}$
S verifies C authentication response $F(pw, realm, P_c, os, \beta, \alpha, P_s)$ with *pw*, and prepares an authentication response $F(pw, realm, P_s, os, \alpha, \beta, P_c)$. Thus, S derives session key $sk = s\beta = scP$. S responds with an appropriate error message or grants access. If it grants access, the responses consist of the username P_c, realm string *realm*, opaque string *os* and the hash value $F(pw, realm, P_s,\ os, \alpha, \beta, P_c)$ based on *pw*.
(5) U verifies S authentication response $F(pw, realm, P_s, os, \alpha, \beta, P_c)$ using the shared secret *pw*.

After mutual authentication between U and S, $sk = csP$ is used as a shared session key.

4.2. Off-line password guessing attacks

A guessing attack involves an attacker – randomly or systematically – trying long-term private keys (e.g., user passwords or server secret keys) one at a time, in a hope of finding the correct private key. Ensuring that long-term private keys are chosen from a sufficiently large space can reduce the risk of exhaustive searches. Most users, however, select passwords from a small subset of the full password space. Such weak passwords with a low entropy are easily guessed by using so-called dictionary attack. For example, one alphanumerical character has 6 bits of entropy, and thus the goal of the attacker, which is to obtain a legitimate communication party's password, can be achieved within a reasonable time. Therefore, the password guessing attacks on Wu et al.'s scheme should be considered as a real possibility.

Wu et al. insisted that their proposed protocol is immune to the off-line password guessing attacks because of the following reasons. *The attacker guesses a password pw and computes* $F(pw)$. *Then, he/she computes* $F(pw, realm, P_c, os, (F(pw) \oplus (F(pw) \oplus \beta)),\ (F(pw) \oplus (F(pw) \oplus \alpha), P_s)$ *and* $F(pw, realm, P_s, os,\ (F(pw) \oplus (F(pw) \oplus \alpha)),\ (F(pw) \oplus (F(pw) \oplus \beta), P_c)$. *Obviously, the attacker cannot compute the password pw to match the RESPONSE, because it faces the difficulty of discrete logarithms*. Unfortunately, this claim is not true. Unlike their claim, these two hash values are not based on the difficulty of discrete logarithms because they do not include the Diffie–Hellman secret key (e.g., $sk = c\beta P = s\alpha P = csP$). From two hash values, we can find out that the attacker's unknown value is only the password *pw*. Therefore, the attacker can easily obtain the password *pw* by performing the following off-line password guessing attacks.

In Steps (2) and (3) of Wu et al.'s scheme, an attacker can collect the following information $\{P_c,\ P_s,\ os,\ \alpha \oplus F(pw),\ \beta \oplus F(pw),\ 0.35emF(pw, realm, P_c, os, \beta, \alpha, P_s)\}$ as a verifiable-text. It is easy to obtain the information since $\{P_c,\ P_s,\ os,\ \alpha \oplus F(pw),\ \beta \oplus F(pw),\ F(pw, realm, P_c, os, \beta, \alpha, P_s)\}$ are readily available over the open network. Upon intercepting $\{P_c,\ P_s,\ os,\ \alpha \oplus F(pw),\ \beta \oplus F(pw),\ F(pw, realm, P_c, os, \beta, \alpha, P_s)\}$, the attacker can perform an off-line password guessing attack as follows.

1. The attacker generates a candidate password pw^* from a password dictionary D and computes $F(pw^*)$.
2. With the knowledge of $P_c,\ P_s,\ os,\ \alpha \oplus F(pw),\ \beta \oplus F(pw),\ F(pw, realm, P_c, os, \beta, \alpha, P_s),\ pw^*$, and $F(pw^*)$, the attacker computes $F(pw^*, realm, P_c, os, \beta \oplus F(pw) \oplus F(pw^*),\ \alpha \oplus F(pw) \oplus F(pw^*), P_s)$ and checks if it is the same as $F(pw, realm, P_c, os, \beta, \alpha, P_s)$. If so, it means that pw^* is the client's real password *pw*. Otherwise, the attacker generates another candidate password pw^{**} and repeats the same step until $F(pw^{**}, realm, P_c, os,\ \beta \oplus F(pw) \oplus F(pw^{**}), \alpha \oplus F(pw) \oplus F(pw^{**}),\ P_s)$ is equal to the client authentication response $F(pw, realm, P_c, os, \beta, \alpha, P_s)$.

In addition, the attacker can also perform the off-line password guessing attacks by using the server authentication response $F(pw, realm, P_s, os, \alpha, \beta, P_c)$ in Step (4). The algorithm of an off-line password guessing attack is given as follows.

```
Guessing Attack (P_c, P_s, os, α ⊕ F(pw), β ⊕ F(pw),
  F(pw, realm, P_c, os, β, α, P_s), D)
{
  for i := 0 to |D|
  {
    pw* ← D;
    fpw* = F(pw*);
    α* = α ⊕ F(pw) ⊕ fpw*;
    β* = β ⊕ F(pw) ⊕ fpw*;
    if F(pw*, realm, P_c, os, β*, α*, P_s) = F(pw, realm, P_c, os,
    β, α, P_s) then
      return pw*
  }
}
```

5. Proposed SIP authentication scheme

This section proposes a new secure and efficient SIP authentication scheme based on elliptic curve cryptography (ECC) in order to overcome the aforementioned security problems with both Durlanik et al.'s and Wu et al.'s authentication schemes. The proposed scheme exploits the key block size, speed, and security jointly. The proposed scheme consists of three phases: the system setup phase, the registration phase, and the authentication phase.

5.1. System setup phase

In this phase, U and S agree on the following system parameters: U and S choose an elliptic curve E over a finite field $GF(q)$. Let $E(GF_q)$ be an additive group of points on an elliptic curve E over a finite field $GF(q)$. Let P be the generating element (point) of $E(GF_q)$.

5.2. Registration phase

When a user client agent U wants to register and become a new legal user, U and S execute the following steps over a secure channel.

(1) $U \rightarrow S$: {username, $F(pw)$}
If U with an identity username and password *pw* wants to

register at the SIP server S, he/she computes $F(pw)$ and sends it with username to S over a secure channel.
(2) S computes $V = F(pw) \oplus F(\text{username}, x)$ and then saves username and V in the verification database table, where x is a secret key of S. Here, the purpose of V is to prevent stolen verifier attacks.

5.3. Authentication phase

Fig. 6 illustrates the proposed SIP authentication scheme and it proceeds as follows.

(1) $U \rightarrow S$: REQUEST (username, $cP \oplus F(pw)$)
U generates a random integer c, computes $cP \oplus F(pw)$, and then sends it with a request message as REQUEST (username, $cP \oplus F(pw)$) to S.
(2) $S \rightarrow C$: CHALLENGE (realm, sP, $F(username, sk)$)
Upon receiving the request message, S derives cP by computing $cP \oplus F(pw) \oplus F(pw)$. Then, S generates a random integer s, and computes a common secret session key $sk = scP$ and a message authentication code F(username, sk). Finally, S sends a challenge message CHALLENGE (realm, sP, F (username, sk)) to U.
(3) $U \rightarrow S$: RESPONSE (username,realm, F(username, realm, sk))
Upon receiving the challenge message, U computes a secret session key $sk = scP$. Then, U computes $F(username, sk)$ and verifies whether it is equal to the received challenge F(username, sk). If they are not equal, U rejects the server challenge message. Otherwise, U authenticates S and computes a message authentication code F(username, realm, sk). Finally, U sends a response message RESPONSE (username, realm, F (username, realm, sk)) to S.
(4) Upon receiving the response message, S computes F(username, realm, sk) and verifies whether it is equal to the received response F(username, realm, sk). If they are not equal, S rejects the user response message. Otherwise, S authenticates U and accepts the user's login request.

After mutual authentication between U and S, $K = csP$ is used as a shared session key.

6. Security analysis

This section provides a security analysis of the proposed SIP authentication scheme.

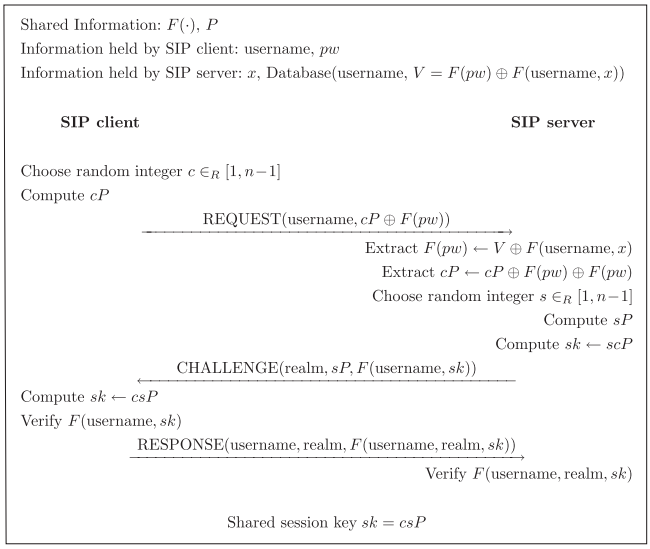

Fig. 6. The proposed SIP authentication scheme.

6.1. Security terms

We define the security terms [21–24] needed for security analysis of the proposed scheme as follows.

Definition 1. A weak secret (Password pw) is a value of low entropy $Weak(k)$, which can be guessed in polynomial time.

Definition 2. A strong secret (Secret x) is a value of high entropy $Strong(k)$, which cannot be guessed in polynomial time.

Definition 3. The Elliptic Curve Discrete Logarithm Problem (ECDLP) is defined as follows. Given a public key point $Q = \alpha P$, it is hard to compute the secret key α.

Definition 4. The Elliptic Curve Diffie–Hellman Problem (ECDHP) is defined as follows. Given point elements αP and βP, it is hard to find $\alpha\beta P$.

Definition 5. A secure one-way hash function $y = F(x)$ is the one when given x it is easy to compute y and given y it is hard to compute x.

6.2. Security properties

The following security properties [21–24]: replay attack, password guessing attack, man-in-the-middle attack, modification attack, Denning-Sacco attack, stolen-verifier attack, mutual authentication, known-key security, session key security, and perfect forward secrecy, must be considered for the proposed SIP authentication scheme.

(1) *Replay attacks*: A replay attack is an offensive action in which an adversary impersonates or deceives another legitimate participant through the reuse of information obtained in a protocol.
(2) *Guessing attacks*: A guessing attack involves an adversary – randomly or systematically – trying long-term private keys (e.g., user passwords or server secret keys) one at a time, in a hope of finding the correct private key. Ensuring that long-term private keys are chosen from a sufficiently large space helps resist against exhaustive searches. Most users, however, select passwords from a small subset of the full password space. Such weak passwords with a low entropy are easily guessed by using so-called dictionary attacks.
(3) *Man-in-the-middle attacks*: The man-in-the-middle attack is a form of active eavesdropping in which the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection where in fact the entire conversation is controlled by the attacker.
(4) *Modification attacks*: A modification attack is an attempt by an adversary to modify information in an unauthorized manner. This is an attack against the integrity of the information.
(5) *Denning-Sacco attacks*: The Dennig-Sacco attack works where an attacker compromises an old session key and tries to find a long-term private key (e.g., user password or server private key) or other session keys.
(6) *Stolen-verifier attacks*: In most applications, the server stores verifiers of users' passwords (e.g., hashed passwords) instead of the clear text of passwords. The stolen-verifier attack means that an adversary who steals the password-

verifier from the server can use it directly to masquerade as a legitimate user in a user authentication process.

(7) *Mutual authentication*: Mutual authentication means that both client and server are authenticated to each other within the same protocol.

(8) *Known-key security*: Known-key security means that each run of an authentication and key agreement scheme between two communication entities (the client and the server) should produce unique secret keys; such keys are called session keys.

(9) *Session key security*: Session key security means that at the end of the key exchange, the session key is not known by anyone but only the two communication entities (the client and the server).

(10) *Perfect forward secrecy*: Perfect forward secrecy means that if a long-term private key (e.g., client password or server private key) is compromised, this does not compromise any earlier session keys.

6.3. Security analysis

With the above definitions, the following theorems are used to analyze the security properties in the proposed SIP authentication scheme.

6.3.1. Replay attacks

The proposed scheme can resist the replay attacks. Suppose an attacker *Eve* intercepts $REQUEST(username, cP \oplus F(pw))$ from U in Step (1) and replays it to impersonate U. However, *Eve* cannot compute a correct session key sk and deliver it to S in Step (3) unless she can correctly guess password pw to obtain cP and guess the right s from sP. When *Eve* tries to guess c from cP or s from sP, she will face the ECDLP. On the other hand, suppose *Eve* intercepts $CHALLENGE(realm, sP, F(username, sk))$ from S in Step (2) and replays it in order to impersonate S. For the same reason, if *Eve* cannot gain the correct c from $cP \oplus F(pw), U$ will find out that F(username, sk) is not equivalent to his/her computed F(username, sk). Then, U will not send RESPONSE (username, realm, F(username, realm, sk)) back to *Eve* in Step (3). Therefore, the proposed scheme can resist against the replay attacks.

6.3.2. Password guessing attacks

The proposed scheme can resist the password guessing attacks. An on-line password guessing attack cannot succeed, since S can choose appropriate trail intervals. On the other hand, in an off-line password guessing attack, *Eve* can try to find a weak password by repeatedly guessing possible passwords and verifying the correctness of the guesses based on information obtained in an off-line manner. In our scheme, *Eve* can gain knowledge of $cP \oplus F(pw)$, sP, F(username, sk) and F(username, realm, sk) in Steps (1), (2), and (3), respectively. In order to obtain the password pw of U, *Eve* first guesses password pw^* and then finds $c^*P = cP \oplus F(pw) \oplus F(pw)$. By using c^*P and sP, *Eve* will try to compute the session key $sk^* = c^*sP$. However, *Eve* has to break the ECDLP and ECDHP to find the keying material $sk^* = c^*sP$ from c^*P and sP to verify her guess. But, *Eve* cannot gain the session key without c^* of c^*P and s of sP because of ECDLP. Therefore, the proposed scheme can resist against the password guessing attacks.

6.3.3. Man-in-the-middle attacks

The proposed scheme can resist against the man-in-the-middle attacks. A mutual password pw between U and S is used to prevent the man-in-the-middle attacks. The illegal attacker *Eve* cannot pretend to be U or S to authenticate since she does not own the mutual password pw. Therefore, the proposed scheme can resist against the man-in-the-middle attacks.

6.3.4. Modification attacks

The proposed scheme can resist against the modification attacks. *Eve* may modify the communication messages $cP \oplus F(pw)$, sP, F(username, sk) and F (username, realm, sk) being transmitted over an insecure network. However, although *Eve* can forge them, the proposed scheme can detect this modification attack, because it can verify not only the equality of $sk = csP$ computed by each party, but also the correctness of $cP \oplus F(pw)$ and sP transmitted between two parties, by validating F(username, sk) and F(username, realm, sk) in the proposed scheme. Therefore, the proposed scheme can resist against the modification attacks.

6.3.5. Denning-Sacco attacks

The proposed scheme can resist against the Denning-Sacco attacks. Although an attacker *Eve* can obtain the fresh session key $sk = csP$, *Eve* cannot obtain the client's secret password pw from $cP \oplus F(pw)$ because *Eve* will face the ECDLP by Definition 2 to obtain c from csP. Therefore, the proposed scheme can resist against the Denning-Sacco attacks.

6.3.6. Stolen-Verifier attacks

The proposed scheme can resist against the stolen-verifier attacks. Servers are always the target of attacks. *Eve* may acquire $V = F(pw) \oplus F(username, x)$ stored in S. However, without knowing S's secret key x, *Eve* cannot forge a login request to pass the authentication, as $F(pw)$ is hidden in $V = F(pw) \oplus F$ (username, x) using S's secret key x, and thus the correctness of the guessed password $F(pw)^*$ cannot be verified by checking $F(pw)^* = F(pw)$. Therefore, the proposed scheme can resist against the stolen-verifier attacks.

6.3.7. Mutual authentication

The proposed scheme provides mutual authentication. The proposed scheme uses the Elliptic Curve Diffie–Hellman key exchange algorithm to provide the mutual authentication. Then the key is explicitly authenticated by a mutual confirmation fresh session key $sk = csP$, where explicit key authentication is the property where both implicit key authentication and key confirmation are satisfied. Therefore, the proposed scheme provides mutual authentication.

6.3.8. Known-key security

The proposed scheme provides known-key security. Knowing a session key $sk = csP$ and the random values c and s is useless for computing the other session keys $sk' = c's'P$, since without knowing c' and s' it is impossible to compute the session key sk'. Therefore, the proposed scheme provides the known-key security.

6.3.9. Session key security

The proposed scheme provides session key security. The session key $sk = csP$ is not known by anyone but only U and S since the random values c and s are protected by the ECDLP, ECDHP, and the secure one-way hash function. Nothing about this session key $sk = csP$ is known to anybody but U and S. Therefore, the proposed scheme provides the session key security.

6.3.10. Perfect forward secrecy

The proposed scheme provides perfect forward secrecy. If the client's password pw and the server's secret key x are compromised, it does not allow an attacker *Eve* to determine the session key sk for the past sessions and decrypt them, since *Eve* still faces the ECDHP to compute the session key $sk = csP$ from the two extracted values cP and sP. Therefore, the proposed scheme satisfies the property of perfect forward secrecy.

The security properties of the previously reported schemes [12,14,19] and the proposed scheme are summarized in Table 2.

Table 2
Comparisons of the security properties of different schemes.

	Yang et al.'s scheme [12]	Durlanik et al.'s scheme [14]	Wu et al.'s scheme [19]	Proposed scheme
Replay attack	Secure	Secure	Secure	Secure
Password guessing attack	Secure	Secure	Insecure	Secure
Man-in-middle attack	Secure	Secure	Secure	Secure
Modification attack	Secure	Secure	Secure	Secure
Denning-Sacco attack	N/A	Insecure	Secure	Secure
Stolen-verifier attack	Insecure	Insecure	Insecure	Secure
Mutual authentication	Provided	Provided	Provided	Provided
Known-key security	N/A	Provided	Provided	Provided
Session key security	N/A	Provided	Provided	Provided
Perfect forward secrecy	N/A	Provided	Provided	Provided

Table 3
Comparisons of computational costs.

	Yang et al.'s scheme [12]	Durlanik et al.'s scheme [14]	Wu et al.'s scheme [19]	Proposed scheme
# of exponentiations	4	0	0 0	0
# of ECC computations	0	4	4	4
# of hash functions	8	8	6	5
# of exclusive-or	4	4	4	3
# of rounds	3	3	4	3
Security	DLP	ECDLP	ECDLP	ECDLP

7. Performance comparisons

The computation costs of the proposed scheme and the previously reported schemes [12,14,19] are shown in Table 3. Generally, the elliptic curve discrete logarithm problem (ECDLP) with an order of 160 bit prime offers approximately the same level of security as the discrete logarithm problem (DLP) with 1024 bit modulus [21].

The proposed SIP authentication scheme requires four ECC multiplications and four hash operations during the protocol execution. Four ECC computations are needed to prevent a Denning-Sacco attack and to provide perfect forward secrecy. When considering hashing and exclusive-or operations, the proposed scheme requires just five hashing operations and three exclusive-or operations for mutual authentication. Obviously, the proposed scheme is more efficient than the previous authentication schemes for Session Initiation Protocol.

8. Conclusions

This paper has revealed the vulnerabilities of both Durlanik et al.'s and Wu et al.'s authentication schemes for session initiation protocol (SIP) to off-line password guessing attacks, Denning-Sacco attacks, and stolen-verifier attacks. In order to resolve those security problems, a new secure and efficient SIP authentication scheme for converged VoIP networks based on elliptic curve cryptosystem (ECC) has been proposed. It has been demonstrated that the proposed SIP authentication scheme resists against those attacks through exploiting the key block size, speed, and security jointly. Through the performance comparisons, we have shown that the proposed scheme is more efficient and preferable for the applications which require low memory and rapid transactions.

Acknowledgements

The authors thank the anonymous reviewers for their constructive comments, which helped us to improve the overall quality of the paper. This research was supported by Ministry of Culture, Sports and Tourism (MCST) and Korea Culture Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program, the Korean government. This research was supported by the Brain Korea 21 program, Ministry of Education, Science and Technology, the Korean government.

References

[1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, SIP: session initiation protocol, IETF RFC3261, June 2002.
[2] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, L. Stewart, HTTP authentication: basic and digest access authentication, IETF RFC2617, June 1999.
[3] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg, SIP: session initiation protocol, IETF RFC2543, March 1999.
[4] M. Thomas, SIP Security Requirements, IETF Internet Draft (draftthomas-sip-sec-reg-00.txt), work in progress, November 2001.
[5] J. Arkko, V. Torvinen, G. Camarillo, A. Niemi, T. Haukka, Security mechanism agreement for SIP sessions, IETF Internet Draft (draft-ietf-sipsec-agree-04.txt), June 2002.
[6] A.B. Johnston, SIP: Understanding the Session Initiation Protocol, second ed., Artech House, 2004.
[7] J.T. Ryu, B.H. Roh, K.Y. Ryu, Detection of SIP flooding attacks based on the upper bound of the possible number of SIP messages, KSII Transactions on Internet and Information Systems (TIIS) 3 (5) (2009) 507–526.
[8] H.N. Yun, S.C. Hong, H.W. Lee, Stateful virtual proxy for sip message flooding attack detection, KSII Transactions on Internet and Information Systems (TIIS) 3 (3) (2009) 251–265.
[9] M. Garcia-Martin, E. Henrikson, D. Mills, Private header (P-Header) extensions to the session initiation protocol (SIP) for the 3rd-generation partnership project(3GPP), IETF RFC3455, 2003.
[10] J.H. Jo, J.S. Cho, Cross-layer optimized vertical handover schemes between mobile Wimax and 3G networks, KSII Transactions on Internet and Information Systems (TIIS) 2 (4) (2008) 171–183.
[11] L. Veltri, S. Salsano, D. Papalilo, SIP security issues: the SIP authentication procedure and its processing load, IEEE Network 16 (6) (2002) 38–44.
[12] C.C. Yang, R.C. Wang, W.T. Liu, Secure authentication scheme for session initiation protocol, Computers and Security 24 (2005) 381–386.
[13] W. Diffie, M. Hellman, New directions in cryptology, IEEE Transaction on Information Theory 22 (6) (1976) 644–654.
[14] A. Durlanik, I. Sogukpinar, SIP authentication scheme using ECDH, World Enformatika Society Transaction on Engineering Computing and Technology 8 (2005) 350–353.
[15] N. Koblitz, Elliptic curve cryptosystems, Mathematics of Computation 48 (1987) 203–209.
[16] V. Miller, Uses of elliptic curves in cryptography, in: Advances in Cryptology CRYPTO'85, Lecture Notes in Computer Science, vol. 218, Springer-Verlag, 1986, pp. 417–426.
[17] NIST, Recommended elliptic curves for federal government use, July 1999.
[18] I. Branovic, R. Giorgi, E. Martinelli, A workload characterization of elliptic curve cryptography methods in embedded environments, ACM SIGARCH Computer Architecture News 32 (3) (2004) 27–34.
[19] L. Wu, Y. Zhang, F. Wang, A new provably secure authentication and key agreement protocol for SIP using ECC, Computer Standards and Interfaces 31 (2) (2009) 286–291.
[20] R. Canetti, H. Krawczyk, Analysis of key-exchange protocols and their use for building secure channels, in: Proc. Eurocrpt 2001, Lecture Notes in Computer Science, 2045, 2001, pp. 453–474.
[21] A.J. Menezes, P.C. Oorschot, S.A. Vanstone, Handbook of Applied Cryptograph, CRC Press, New York, 1997.
[22] D. Denning, G. Sacco, Timestamps in key distribution systems, Communications of the ACM 24 (1981) 533–536.
[23] C.L. Lin, T. Hwang, A password authentication scheme with secure password updating, Computers and Security 22 (1) (2003) 68–72.
[24] E.J. Yoon, W.H. Kim, K.Y. Yoo, Robust and simple authentication protocol for secure communication on the web, in: ICWE 2005, Lecture Notes in Computer Science, vol. 3579, Springer-Verlag, 2005, pp. 352–362.