Secure and Efficient SIP Authentication Scheme for Converged VoIP Networks

Qiong Pu[1,2] and Shuhua Wu[3]
[1] CIMS Research Center, Tongji Univerity, China
[2] State Key Laboratory of Information Security, Graduate University of Chinese Academy of Sciences, China
3 Department of Network Engineering, Information Engineering University, China

Abstract: *The Session Initiation Protocol (SIP) is commonly used to establish Voice over IP (VoIP) calls. Mostly recently, Yoon et al. [27] proposed a new secure and efficient SIP authentication scheme in a converged VoIP network based on Elliptic Curve Cryptosystem (ECC). In this paper, we first demonstrate that the recently proposed SIP authentication scheme is insecure against off-line password guessing attacks. Thereafter, we propose an enhanced SIP authentication scheme that enjoys provable security. And yet our scheme is simple and efficient. Therefore, the end result is more suited to be a candidate for SIP authentication scheme.*

Keywords: *VoIP, SIP, elliptic curve, authentication.*

Received August 30, 2010; accepted October 24, 2010

1. Introduction

Voice over Internet Protocol (VoIP) is a fast growing technology believed to be the future replacement for traditional Public Switched Telephone Network (PSTN). There are many protocols used in VoIP signaling, but Session Initiation Protocol (SIP) is one of the widely used ones. It has been chosen by the Third-Generation Partnership Project (3GPP) as the protocol for multimedia application in 3G mobile networks. The SIP [11, 21] is an application-layer protocol that is capable of handling all the signalling requirements of a VoIP session, i.e., initiating, managing and terminating voice and video sessions across packet networks. It is analogous to the SS7 [13] protocol in traditional telephony. Security and privacy requirements in a VoIP environment are expected to be equivalent to those in PSTN.

However, the original authentication scheme for SIP-based service typically uses HTTP digest authentication protocol [10], which was found vulnerable to the off-line password guessing attacks and the server spoofing attacks [26]. Then Yang *et al.* [26] proposed an SIP authentication scheme but it is not suitable for devices with a low computational power because it works only for Discrete Logarithm (DL) settings and involves in costly exponential computation. 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. Therefore, the services that are enabled by SIP should be equally applicable to mobile and ubiquitous computing [27]. To meet this goal, based on Yang *et al.* scheme, Durlanik *et al.* [9] and Wu *et al.* [24] independently proposed an efficient SIP authentication schemes using Elliptic Curve Cryptosystem (ECC), which has the well-known advantages with regard to processing and size constraints [12, 14]. Mostly recently, Yoon *et al.* [27] pointed out that both Durlanik *et al.* and Wu *et al.* SIP authentication schemes are still vulnerable to off-line password guessing attacks, Denning-Sacco attacks, and stolen-verifier attacks [7, 15, 16] and they then proposed a new SIP authentication scheme in a converged VoIP network based on ECC in order to overcome those security problems.

In this paper, we demonstrate Yoon *et al.* scheme [27] is still vulnerable to off-line password guessing attacks. Thereafter, we propose a practical SIP authentication scheme using ECC. Our scheme is quite efficient and is suitable for the user's device with limited computing capability. And it is quite simple to use since it also constructs authentication with easily-remembered passwords only. To use the SIP service, the user just needs to know his identity and password. However, many previous SIP authentication schemes, e.g., [18, 20, 22, 23, 24] accomplish authentication using long, high-entropy cryptographic keys, which is inconvenient for human users. Furthermore, we can prove our scheme is secure against dictionary attacks. Even when the verifier stored in the server is stolen, the attacker can not guess the correct password. That is to say our scheme can resist against the stolen-verifier attacks. Therefore, the end result is more suited to be a candidate for SIP authentication scheme.

The remainder of this paper is organized as follows. Section 2 reviews SIP architecture and its authentication scheme. Section 3 briefly reviews Yoon *et al.* SIP authentication scheme and demonstrates its weaknesses. Section 4 provides an improved scheme to overcome all those disadvantages existing in the previous scheme. In addition, some important discussions are also made in this section. Section 5 provides the rigorous proof of the security for our scheme. Finally, conclusions is presented in section 6.

2. SIP Authentication

In this section, we briefly review SIP architecture and SIP authentication scheme.

2.1. SIP Architecture

SIP is a call setup signaling protocol for IP-based telephony services. The SIP architecture is mainly composed of a proxy server, redirect server, user agent, register server, and location server. The function of each component is described as follows. Here we follow the description in [26, 27].

- *User Agent:* A user agent is a logical entity, such as a callee or a caller.
- *Proxy Server:* A proxy server forwards a request and response between a callee and caller. 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 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.

The main signaling "services" of the SIP protocol are:

1. The establishment.
2. The cancellation.
3. The termination of a multimedia or voice session among two or more participants.

The corresponding SIP messages are: INVITE, CANCEL, and BYE. As in [8], we consider the case where a user A (caller) wishes to establish a multimedia connection with user B (callee). The caller generates an INVITE message and sends it to the corresponding proxy, which in turn forwards it to the callee. Assuming that the calee is available the session is established. When either of the participants wishes to terminate the session he must issue a BYE message. The establishment-termination process is depicted in Figure 1 [8].

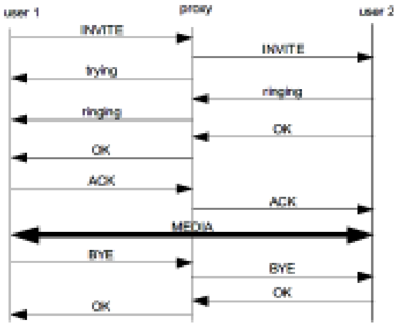

Figure 1. SIP establishment and termination procedure.

2.2. SIP Authentication

SIP is based on the application-layer and is a text-based client-server protocol. When a user requests to use an SIP service, he needs to be authenticated first before getting the service from the server. In SIP specification [21], the authentication mechanism proposed is HTTP digest based authentication. In SIP terms, HTTP digest mechanism is called the SIP authentication. Next we follow the description [26] to introduce it. HTTP Digest authentication [10] is a challenge-based mechanism. Before the scheme starts, the client pre-shares a password with the server. Note that the pre-share password is used to verify the identity of the client or the server because only these two sides have the pre-share password. The protocol proceeds as follows, as depicted in Figure 2.

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

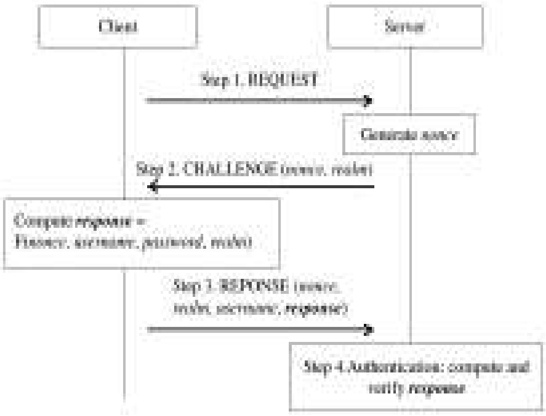
Client
Server
Step 1. REQUEST
Generate nonce
Step 2. CHALLENGE (nonce, realm)
Compute *response* =
F(nonce, username, password, realm)
Step 3. REPONSE (nonce, realm, username, *response*)
Step 4 Authentication: compute and verify *response*

Figure 2. Digest authentication.

Please note, HTTP digest authentication protocol is found failing to provide security at an acceptable level and vulnerable to many attacks such as offline password guessing attack, server spoofing attack [26]. For this reason, it is very important to propose a practical SIP authentication scheme to overcome these shortcomings.

3. Review of Yoon et al.'s Scheme

This section describes the SIP authentication scheme proposed by Yoon *et al.* [27], starting with some notations.

3.1. Notations

The notations used in their scheme are described as the following:

- *U:* The user.
- *S:* The server.
- *D:* A uniformly distributed dictionary of size |*D*|.
- *pw:* A low-entropy password of *U* which is randomly chosen from *D*.
- *E:* An elliptic curve defined over a finite field F_p with large group order [12, 14].
- *P:* A point in *E* with large order q, where q is a secure large prime.
- *G:* The cyclic addition group generated by *P*.
- *sP:* The point multiplication defined as $sP = \underbrace{P + P + \cdots + P}_{s\text{times}}$, where *s* is an integer in Z_q.
- $\oplus$ *:* A bit-wise exclusive-or (XOR) operation.
- *F(.):* is a one-way hash function $F: \{0,1\}^* \rightarrow \{0,1\}^l$, where *l* is the security parameter.

In an Elliptic Curve Cryptosystem (ECC), the elliptic curve equation is usually defined as the form of $E: y^2 = x^3 + ax + b (\bmod p)$ over a prime finite field F_p, where $a,b \in F_p$, $p>3$, and $4a^3 + 27b^2 \neq 0 (\bmod p)$ [12]. We look at the points on *E* with coordinates in F_p. Then the points on *E* together with the extra point living "at infinity" *O*, which we denote by $E_p(a,b)=\{(x,y):x,y \in F_p$ satisfy $y^2=x^3+ax+b(mod\ P)\} \cup \{O\}$, obey the elliptic curve addition algorithm and thus form an additive group. In general, the group *G* is a subgroup of $E_p(a,b)$. In view of shortness, we omit the details and refer to [12, 14].

3.2. Description

Yoon *et al.'s* scheme consists of three phases: the system setup phase, the enrollment phase, and the authentication phase. Here, we just follow the description in [27].

- *System Setup Phase: S* generates and publishes the following system parameters *(G,q,P,F). U* must agree on these system parameters.
- *Enrollment Phase:* If *U* with an identity username and password *pw* wants to register at the SIP server *S* and become a new legal user, he/she computes *F(pw)* and sends (username, *F(pw)*) to *S* over a secure channel. Then *S* computes $V = F(pw) \oplus F(\text{username}, k)$ and saves (username, *V*) in the verification database table, where *k* is a secret key of *S*. Here, the purpose of *V* is to prevent stolen verifier attacks.
- *Authentication Phase:* Figure 3 [27] illustrates Yoon *et al.'s* SIP authentication scheme and it proceeds as follows:
  1. *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. Upon receiving the request message, *S* first extracts *F(p*w) by computing $V \oplus F(username, k)$ with his private key *k* and 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. 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*, *sk=csP* is used as a shared session key.

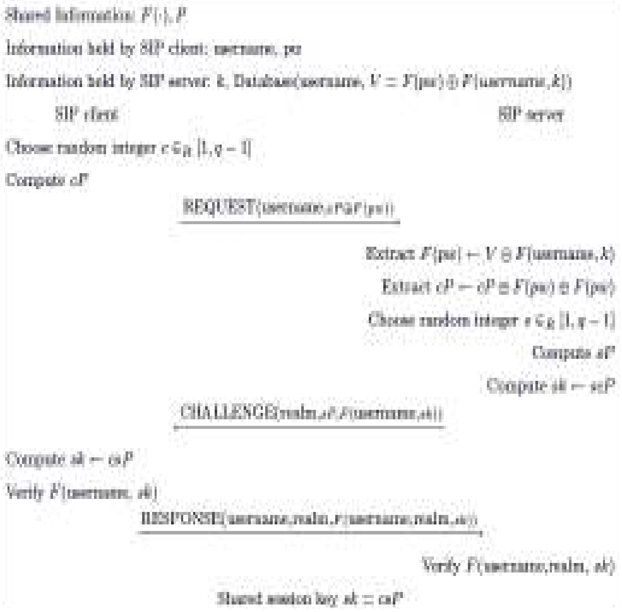
Shared Information: $F(\cdot)$, P
Information held by SIP client: username, pw
Information held by SIP server: k, Database(username, $V = F(pw) \oplus F(username, k)$)
SIP client
SIP server
Choose random integer $c \in_R [1, q-1]$
Compute cP
REQUEST(username, $cP \oplus F(pw)$)
Extract $F(pw) \leftarrow V \oplus F(username, k)$
Extract $cP \leftarrow cP \oplus F(pw) \oplus F(pw)$
Choose random integer $s \in_R [1, q-1]$
Compute sP
Compute $sk \leftarrow scP$
CHALLENGE(realm, sP, $F(username, sk)$)
Compute $sk \leftarrow csP$
Verify $F(username, sk)$
RESPONSE(username, realm, $F(username, realm, sk)$)
Verify $F(username, realm, sk)$
Shared session key $sk = csP$

Figure 3. Yoon et al.'s SIP authentication scheme.

3.3. Weaknesses of Yoon et al's Scheme

Unfortunately, Yoon *et al.'s* [27] scheme described above is completely insecure. In this section, we will show it is vulnerable to an off-line password guessing attack. Our attack is inspired by the work of [5]. Off-line password guessing attack succeeds when there is information in communications, which can be used to verify the correctness of the guessed passwords. In [27], Yoon et al. claimed that their scheme can resist the off-line password guessing attack. However, we can show that the off-line password guessing attack, not as they claimed, is still effective in Yoon et al.'s scheme. In Yoon et al.'s scheme, since all transcripts are transmitted over an open network, a benign (passive) adversary, can easily obtain the valid message transcript of $cP \oplus F(pw)$. The adversary can guess a password *pw** from D and derive the corresponding $(x^*, y^*) = cP \oplus F(pw) \oplus F(pw^*)$, then verify it by checking $(y^*)^2 \overset{?}{=} (x^*)^3 + ax^* + b (\bmod p)$. Clearly, if *pw** is not correct, the computation $cP \oplus F(pw) \oplus F(pw^*)$ should result in a random pair *(x*, y*)*. Even if $x^*, y^* \in F_p$, the probability that the point *(x*, y*)* lies on *E* is no larger than $\frac{2}{p}$. Typically |*D*| is much less than *p*. Therefore, the adversary should be able to identify the correct password *pw* given one valid message transcript of $cP \oplus F(pw)$ using such a dictionary attack, which a probability of $(1-\frac{2}{p})^{|D|-1} \approx 1-\frac{2(|D|-1)}{p} \approx 1$. Please note, the attack is a brute-force method in essence, i.e., the attacker tries offline all possible passwords in a given small set of values. Even though such attacks are not very effective in the case of high-entropy keys, they can be very damaging when the secret key is a password since the attacker has a non-negligible chance of winning [3].

In addition, we point out that the key derivation phase is deliberately omitted in Yoon et al.'s scheme. Key derivation refers to the process by which an agreed upon large random number, often named master secret, is used to derive session keys to encrypt and authenticate data. As a result, an adversary can obtain some information about the session key although an adversary is unable to obtain the whole key. More specifically, it can reliably distinguish between the session key sk and a randomly chosen string of the expected length simply by checking if sk is a point on elliptic curve or not. In some sense, this is another weakness of the scheme. Indeed, the key derivation phase is a crucial step for theoretical reasons, but also practical purpose, and can not be omitted.

Finally, it is worth of noticing that similar attacks can be applicable to Durlanik *et al.'s* scheme [9] and Wu et al.'s authentication scheme [24]. Please note Yoon et al's did not find such weakness in [27]. Since the rationale for it is the same with the attack described above, the description is omitted here.

4. Our Proposed SIP Authentication Scheme

This section proposes a new secure and efficient SIP authentication scheme using ECC in order to overcome the aforementioned security problems with Yoon et al.'s authentication scheme. Our scheme is based on the password-based protocol in [6].

4.1. Description

Our scheme also consists of three phases: the system setup phase, the enrollment phase, and the authentication phase. And the first two are quite similar to Yoon *et al.'s* except with *V=G(pw)+G(username, k)*, where $G: \{0,1\}^* \rightarrow G$ is a full domain hash function and also a public parameter. Next, we just give a full description of the authentication phase.

- *Authentication Phase:* Figure 4 illustrates our SIP authentication scheme and it proceeds as follows:

1. *U* generates a random integer *c*, computes *X=cP+G(pw)*, and then sends it with a request message as REQUEST(username, *X*) to *S*.
2. Upon receiving the request message, *S* extracts *G(pw)* by computing *V-G*(username, *k*) and derives *cP* by computing *X-G(pw)*. Then, *S* generates a random integer *s*, and computes *Y=sP, Z=scP* and a message authentication code

α *←F("1",username,realm,X,Y,Z)*. Finally, *S* sends a challenge message CHALLENGE(realm, *Y*, α) to *U*.

3. Upon receiving the challenge message, *U* computes *Z=cY*. Then, *U* computes *F("1",username,realm,X,Y,Z)* and verifies whether it is equal to the received challenge α. If they are not equal, *U* rejects the server challenge message. Otherwise, *U* authenticates *S* and computes a message authentication code β *←F("2",username,realm,X,Y,Z)*. Finally, *U* sends a response message RESPONSE(username, realm, β) to *S*.
4. Upon receiving the response message, *S* computes *F("2",username,realm,X,Y,Z)* and verifies whether it is equal to the received response β. 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, *U* and *S* compute *sk=F("0",username,realm,X,Y,Z)* as the shared session key.

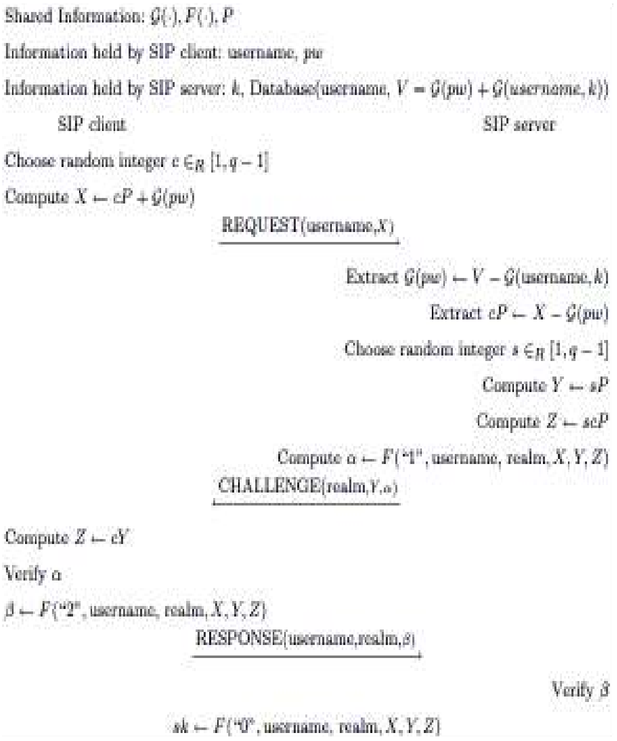
Shared Information: $\mathcal{G}(\cdot), F(\cdot), P$
Information held by SIP client: username, pw
Information held by SIP server: k, Database(username, $V = \mathcal{G}(pw) + \mathcal{G}(username, k)$)
SIP client
SIP server
Choose random integer $c \in_R [1, q-1]$
Compute $X \leftarrow cP + \mathcal{G}(pw)$
REQUEST(username,X)
Extract $\mathcal{G}(pw) \leftarrow V - \mathcal{G}(\text{username}, k)$
Extract $cP \leftarrow X - \mathcal{G}(pw)$
Choose random integer $s \in_R [1, q-1]$
Compute $Y \leftarrow sP$
Compute $Z \leftarrow scP$
Compute $\alpha \leftarrow F(\text{"1"}, \text{username, realm}, X, Y, Z)$
CHALLENGE(realm,Y,α)
Compute $Z \leftarrow cY$
Verify α
$\beta \leftarrow F(\text{"2"}, \text{username, realm}, X, Y, Z)$
RESPONSE(username,realm,β)
Verify β
$sk \leftarrow F(\text{"0"}, \text{username, realm}, X, Y, Z)$

Figure 4. Our SIP authentication scheme.

Now the weakness for helping guess the password in section 3.3 is not available to the intruder since the computation *X-G(pw*)* a results in a point in *E* even if the guessed password *pw** is not correct. Actually, in Section 5, we can prove our scheme can resist any offline password guessing attack. In addition, as in [27], the server in our scheme also stores the user's verifier *V* rather than the user's bare password *pw* in order to reduce the security breach once the server is compromised. Even when the adversary has acquired *V=G(pw)+G(username,k)* stored in S. However, without knowing *S*'s secret key *k*, she cannot forge a login request to pass the authentication, as *G(pw)* is hidden in *V* using *S*'s secret key *k*, and thus the correctness of the guessed password *G(pw)* cannot be verified straight. Therefore, the proposed scheme can resist against the stolen-verifier attacks. This is a quite attractive feature because numerous customers' secrets are stored in the server's databases and the server is always the targets of attackers.

In addition, we can also use puzzle protection to defeat the so-called denial of service attacks. More information about it is referred to [17].

4.2. Security

Here we just provide the intuitive understanding the security for our scheme. And the rigorous proof of the security can be found in next section.

At first, our scheme can provide mutual authentication and establish a secret session key only known by the two communicating parties. Moreover, it can protect the password information against the notorious password guessing attacks by which attackers could search the relatively small space of human-memorable passwords. In other words, it is a secure password-based key agreement protocol with mutual authentication. Next we come to explain it. If the adversary tries to impersonate *S* to *U*, he has to guess the authenticator α and tries to send a corrector one, which is reduced to online guessing the password *pw* since each authenticator sent by the adversary has been computed with at most one password. If the adversary tries to impersonate *U* to *S*, he has to guess the correct password in order to send such an *X* that he can know *c* exactly and thus compute *Z=cy*, which is certainly reduced to online guessing the password against the user. Otherwise, the attacker can not know the real value of *c* (due to the hardness of EC discrete logarithm problem). One can remark that *Y* is generated by *S* in this case and the value of *s* is also unknown to the attacker. As a result, he can not compute *Z* and thus will not be able to compute the valid authenticator β. Therefore, the server will know it is an illegal user and will halt the processing. If the adversary mounts a passive attack, one can also know that the attacker can know nothing about session key *sk* yet since either *c* or *s* is unknown to him and he can not compute Z either. Therefore, an exhaustive on-line attack is the "best" possible strategy for an attacker. However, one can invalidate or block the use of a password whenever a certain number of failed attempts occur. In a word, our scheme is a secure password-based protocol. In addition, with the same analysis, even when *pw* is compromised, the adversary can not know the previous session keys that were established before the corruption which is usually called forward security since the session of this type must involve with both legal user and server.

4.3. Performance

Our scheme is efficient. One can easily remark that the communication cost remains unchanged in terms of rounds when compared with previous schemes since the communication flows are also consistent with that of the standard SIP protocol. And the details of comparisons between our scheme and the existing password-only authentication schemes for SIP so far as I know are shown in Table 1. Note that we measure the computation cost only by the number of point multiplication (resp. modular exponentiation) operations, which entail the highest computational complexity, and neglect the computational complexity of all other operations such as Hash computation, which can be done efficiently. As shown in Table 1, in one run of our scheme, each participant performs only two point multiplications of elliptic curve. Obviously, our scheme is much more efficient than the scheme in [26] because the latter involves in costly exponential computation. When compared with other previous works [9, 24, 27], our scheme is equally efficient and yet provide more security guarantees. Based on the results listed in the table, we conclude that our scheme is more practical than the related authentication schemes for SIP.

Table 1. Comparisons with related works.

Schemes	Security Properties		Computation Cost*
	Password-Guessing Attack	Stolen-Verifier Attack	
Yang's Scheme[26]	Secure	Insecure	2 EXP
Durlanik's Scheme[9]	Insecure	Insecure	2 PM
Wu's Scheme[24]	Insecure	Insecure	2 PM
Yoon's Scheme[27]	Insecure	Secure	2 PM
Our Scheme	Secure	Secure	2 PM

* PM: Elliptic curve point multiplication; EXP: Modular exponentiation.

Based on the software implementation results in [25], we get the running time for both sides in our scheme with p=512bits and a large prime order q=160bits: the client side roughly needs 0.26s if the Philips HiPersmart card is used and the server side needs 2.34ms if Pentium IV processor is used. Therefore, our scheme is Converged VoIP Networks.

5. Security Proof for our Scheme

In this section, we show that our scheme is secure in the random-oracle (ideal hash function) model, starting with the formal security models and some algorithm assumption that will be used in our proof.

5.1. Security Model

In this section, we introduce the formal security models which will be used in next section when we show that our scheme is secure in the random-oracle model. The model is that of Bellare *et al.* [4]. Here we just follow the description in [2, 3].

5.1.1. Protocol Syntax

- *Protocol participants and Long-lived keys:* Each participant in authenticated key agreement is either a client (User) $U \in \mathrm{U}$ or a server $S \in \mathrm{S}$. Each of them may have several instances called oracles involved in distinct, possibly concurrent, executions of the protocol. We denote U (resp. S) instances by U^i (resp. S^j) or by I when we consider any participant instance. And each client U pre-shares a password pw with the server S, where pw is uniformly drawn from the dictionary D.
- *Partner:* An instances is said to be partner of another instance if it has accepted with the same session identifier SID as the latter's, where SID is defined as the concatenation of all messages an instance has sent and received.

5.1.2. Communication Model

The authenticated key agreement protocol P is an interactive algorithm between U^i and S^j that provides the instances of the two communicating parties with a session key sk. The interaction between an adversary A and the protocol participants occurs only via oracle queries, which model the adversary capabilities in a real attack. The types of oracles available to the adversary are as follows:

- *Execute(U^i, S^j):* This query models passive attacks in which the attacker eavesdrops on honest executions between a user instance U^i and a server instance S^j. The output of this query consists of the messages that were exchanged during the honest execution of the protocol.
- *Send(I^i, m):* This query models an active attack, in which the adversary may intercept a message and then either modify it, create a new one, or simply forward it to the intended participant. The output of this query is the message that the participant instance I^i would generate upon receipt of message m.
- *Reveal(I^i):* This query models the misuse of the session key by instance I^i (known-key attacks). If a session key is not defined for instance I^i then return $\perp$. Otherwise, return the session key held by the instance I^i.
- *Corrupt(U):* This query returns to the adversary the long-lived key for participant U, i.e., its password pw. As in [4], we assume the weak corruption model in which the internal states of all instances of that

user are not returned to the adversary.

Obviously, the adversary is in complete control of every aspect of all communications between participants in the model.

5.1.3. Security Definitions

The security notions take place in the context of executing *P* in the presence of the adversary *A*. One first initializes the system parameters, chooses a password from *D* for each user, then provides coin tosses to *A*, all oracles, and runs the adversary by letting it ask any number of queries as described above, in any order.

- *Forward Security:* In order to model the Forward Security (FS) of the session key, we consider the game in which an additional oracle is made available to the adversary: the Test (I^i) oracle. The *Test*-query can be asked at most once by the adversary *A* and is only available to *A* if the attacked instance I^i is FS-Fresh, which is defined to avoid cases in which adversary can trivially break the security of the scheme. In this setting, we say that a session key *sk* is FS-Fresh if all of the following hold:

1. I^i has accepted.
2. No *Corrupt*-query on *I* has been asked since the beginning of the game.
3. No *Reveal*-query has been asked to I^i or to its partner (defined according to the session identification).

In other words, the adversary can only ask *Test*-queries to instances which had accepted before the *Corrupt* query on the related user is asked. This query is answered as follows:

- *Test(I^i):* If no session key for instance I^i is defined, then return the undefined symbol $\perp$. Otherwise, flip a (private) coin *b* and return the session key for instance I^i if *b=1* or a random of key of the same size if *b=0*.

When playing this game, the goal of the adversary is to guess the bit *b* involved in the *Test*-query, by outputting this guess *b'*. We denote *Pr[b=b']* as the probability that *A* correctly guesses the value of *b*. Thus we define *A*'s advantage in breaking the semantic security with regard to *P* as $Adv_{\mathrm{P,D}}^{ake-fs}(\mathrm{A}) = 2Pr[b=b'] - 1$. The protocol *P* is said to be *(t,ε)*-FS-secure if *A*'s advantage is smaller than ε for any adversary *A* running with time *t*. The definition of time-complexity that we use henceforth is the usual one, which includes the maximum of all execution times in the games defining the security plus the code size [1].

Usually, we say a protocol is secure if ε can be negligible (in the security parameter *l*). However, to prevent dictionary attack, ε is just required to be $O(n_{active}/|D|)+\eta(l)$ for password-based protocols, where $|D|$ is the size of the dictionary *D*, n_{active} is the number of active attacks and $\eta(l)$ is a negligible function depending on the security parameter *l*.

5.2. Diffie-Hellman Assumptions

In this subsection, we recall the computational assumptions upon which the security of our protocol is based upon. Here we follow the description in [2]. A $(t,\varepsilon)-ECCDH_{P,G}$ attacker is a probabilistic machine Δ running in time *t* such that its success probability $Succ_{P,G}^{cdh}(A)$, given random elements *uP* and *vP* to output *uvP* (denoted by $ECCDH_{P,G}(uP,vP)$), is greater than ε:

$$Succ_{P,G}^{cdh}(A) = Pr[\Delta(uP,vP) = uvP] \geq \varepsilon.$$

We denote by $Succ_{P,G}^{cdh}(t)$ the maximal success probability over every adversaries running within time *t*. The CDH-Assumption states that $Succ_{P,G}^{cdh}(t) \leq \varepsilon$ for any t/ε not too large. A $(t,n,\varepsilon)-ECGDH_{P,G}$ attacker is a *(t,ε)-ECCDH$_{P,G}$* attacker, with access to an additional oracle: a DDH-oracle, which on any input *(uP,vP,zP)* answers whether $z = uv \bmod q$. Its number of queries is limited to *n*. As usual, we denote by $Succ_{P,G}^{gdh}(n,t)$ the maximal success probability over every such adversaries running within time *t*. The GDH-Assumption states that $Succ_{P,G}^{gdh}(n,t) \leq \varepsilon$ for any t/ε not too large [19].

5.3. Security Proof

As Theorem 1 states, our scheme is secure in the random oracle model as long as we believe that the GDH problem is hard in *G*.

- *Theorem 1:* Let *D* be a uniformly distributed dictionary of size |*D*|. Let *P* describe the password-based authenticated key exchange protocol associated with these primitives as defined in Figure 4. Then, for any adversary *A* within a time bound *t*, with less than q_s active interactions with the parties (*Send*-queries) and q_p passive eavesdroppings (*Execute*-queries), and asking q_f(resp. q_g) hash queries to any *F*(resp. *G*) respectively,

$$Adv_{P,D}^{ake\ \ fs}(A) \leqslant \frac{(q_p + q_s)^2}{q} + \frac{q_f^2}{2^l} 4$$

$$Succ_{P,G}^{gdh}(q_h, t + q_g + 2\tau) + \frac{6q_s}{|D|} + \frac{4q_s}{2^l},$$

where τ represents the computational time for a point multiplication in G. The complete proof is omitted here.

6. Conclusions

In this paper, we have demonstrated that the recently proposed SIP authentication scheme is insecure against off-line password guessing attacks. Thereafter, we have proposed a provable secure SIP authentication scheme using ECC. Our scheme is simple and efficient. Therefore, the end result is more suited to be a candidate for SIP authentication scheme.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (No. 61101112) and China Postdoctoral Science Foundation (No. 2011M500775).

References

[1] Abdalla M., Bellare M., and Rogaway P., "The Oracle Diffie-Hellman Assumptions and an Analysis of DHIES," *in Proceedings of The Cryptographers Track at RSA*, USA, pp. 143-158, 2001.

[2] Abdalla M., Chevassut O., and Pointcheval D., "One-Time Verifier-Based Encrypted Key Exchange," *in Proceedings of Public Key Cryptography, Lectures Notes in Computer Science*, New York, pp.47-64, 2005.

[3] Abdalla M. and Pointcheval D., "Simple Password-Based Encrypted Key Exchange Protocols," *in Proceedings of Topics in Cryptology, Cryptographers Track at RSA, Lectures Notes in Computer Science*, Berlin, pp. 191-208, 2005.

[4] Bellare M., Pointcheval D., and Rogaway P., "Authenticated Key Exchange Secure Against Dictionary Attacks," *in Proceedings of EUROCRYPT, Lectures Notes in Computer Science*, Berlin, pp.139-155, 2000.

[5] Boyd C., Montague P., and Nguyen K., "Elliptic Curve Based Password Authenticated Key Exchange Protocols," *in Proceedings of 6th Australasian Conference Information Security and Privacy*, Berlin, pp. 487-501, 2001.

[6] Bresson E., Chevassut O., and Pointcheval D., "New Security Results on Encrypted Key Exchange," *in Proceedings of 7th International Workshop on Theory and Practice in Public Key Cryptograph*y, New York, pp. 145-158, 2004.

[7] Denning D. and Sacco G., "Timestamps in Key Distribution Systems," *Communications of the ACM*, vol. 24, no. 8, pp. 533-536, 1981.

[8] Dimitris G. and Costas L., "A Lightweight Protection Mechanism Against Signaling Attacks in a SIP-Based VoIP Environment," *Telecommunication Systems*, vol. 36, no. 4, pp. 153-159, 2007.

[9] Durlanik A. and Sogukpinar I., "SIP Authentication Scheme using ECDH," *in Proceedings of World Enformatika Socity Transaction on Engineering Computing and Technology*, pp. 350-353, 2005.

[10] Franks J., Hallam-Baker P., Hostetler J., Lawrence S., Leach P., Luotonen A., and Stewart L., "HTTP Authentication: Basic and Digest Access Authentication," *Technical Document Internet Engineering Task Force RFC*, 1999.

[11] Handley M., Schooler E., Schulzrinne H., and Rosenberg J., "SIP: Session Initiation Protocol," *Technical Document Internet Engineering Task Force RFC*, 1999.

[12] Hankerson D., Menezes A., and Vanstone S., *Guide to Elliptic Curve Cryptography*, Springer-Verlag, USA, 2004.

[13] International Telecommunications Union. "ITU-T Recommendation Q.700: Introduction to CCITT Signalling System 7," Recommendation Q.700, International Telecommunications Union, 1993.

[14] Koblitz N., "Elliptic Curve Cryptosystem," *Mathematics of Computation*, vol. 48, no. 177, pp. 203-209, 1987.

[15] Menezes A., Oorschot P., and Vanstone S., *Handbook of Applied Cryptograph*, CRC Press, New York, 1997.

[16] Lin C. and Hwang T., "A Password Authentication Scheme with Secure Password Updating," *Computers and Security*, vol. 22, no. 1, pp. 68-72, 2003.

[17] Laurens V., Saddik A., and Nayak A., "Requirements for Client Puzzles to Defeat the Denial of Service and the Distributed Denial of Service Attacks," *The International Arab Journal of Information Technology*, vol. 3, no. 4, pp. 326-333, 2006.

[18] Liao Y. and Wang S., "A New Secure Password Authenticated Key Agreement Scheme for SIP using Self-Certified Public Keys on Elliptic Curves," *in Proceedings of Computer Communications*, pp. 372-380, 2010.

[19] Okamoto T. and Pointcheval D., "The Gap-Problems: a New Class of Problems for the Security of Cryptographic Schemes," *in Proceedings of Public Key Cryptography, Lectures Notes in Computer Science*, Berlin, pp. 104-118, 2001.

[20] Ring J., Choo K., Foo E., and Looi M., "A New Authentication Mechanism and Key Agreement Protocol for SIP using Identity-Based Cryptography," *in Proceedings of AusCERT Asia Pacific Information Technology Security*

Conferencet, Australia, pp. 61-72, 2006.

[21] Rosenberg J., Handley M., Schooler E., Sparks R., Peterson J., Johnston A., Camarillo G., and Schulzrinne H., "SIP: Session Initiation Protocol," *Technical Document Internet Engineering Task Force RFC*, 2002.

[22] Vesterinen P., "User Authentication in SIP," *Technical Document TKK T-110.5290 Seminar on Network Security*, 2006.

[23] Wang F. and Zhang Y., "A New Provably Secure Authentication and Key Agreement Mechanism for SIP using Certificateless Public-Key Cryptography," *Computer Communication*, vol. 31, no. 10, pp. 2142-2149, 2008.

[24] Wu L., Wang F., and Zhang Y., "A New Provably Secure Authentication and Key Agreement Protocol for SIP using ECC," *Computer Standard & Interfaces*, vol. 31, no. 2, pp. 286-291, 2009.

[25] Wu T. and Tseng Y., "An Efficient user Authentication and Key Exchange Protocol for Mobile Client-Server Environment," *Computer Networks*, vol. 54, no. 9, pp. 1520-1530, 2009.

[26] Yang C., Wang R., and Liu W., "Secure Authentication Scheme for Session Initiation Protocol," *Computer and Security*, vol. 24, no. 5, pp. 381-386, 2005.

[27] Yoon E., Yoo K., Kim C., Hong Y., Jo M., and Chen H., "A Secure and Efficient SIP Authentication Scheme for Converged VoIP Networks," *Computer Communications*, vol. 33, no. 14, pp. 1674-1681, 2010.

Qiong Pu is a lectuer of Electronics Department, Science Institute, Information Engineering University, China. Now, she is pursuing doctoral degree in school of Electronics and Information Engineering, Tongji University, China.

Shuhua Wu is a lectuer of Networks Engineering Department, Information Science Technology Institute, Zhengzhou, China. His research interests include cryptology and communication protocols.