ISSN 1392 – 124X, ISSN 2335 – 884X (online) INFORMATION TECHNOLOGY AND CONTROL, 2013, Vol.42, No.4

An Enhanced Authenticated Key Agreement for Session Initiation Protocol

Mohammad Sabzinejad Farash[1], Mahmoud Ahmadian Attari[2]

[1] *Department of Information and Communication Technology, Malek Ashtar University of Technology, Tehran, Iran.*
e-mail: sabzinejad@khu.ac.ir

[2] *Faculty of Electrical and Computer Engineering, K. N. Toosi University of Technology, Tehran, Iran*
e-mail: mahmoud@eetd.kntu.ac.ir

crossref http://dx.doi.org/10.5755/j01.itc.42.4.2496

Abstract. In 2012, Xie proposed an authentication scheme based on Elliptic Curve Cryptography (ECC) for Session Initiation Protocol (SIP). However, this paper demonstrates that the Xie's scheme is vulnerable to impersonation at-tack by which an active adversary can easily forge the server's identity. Based on this attack, we also show that the Xie's scheme is also defenceless to off-line password guessing attack. Therefore, we propose a more secure and efficient scheme, which does not only cover all the security flaws and weaknesses of related previous protocols, but also provides more functionalities. We also evaluate the proposed protocol by AVISPA (Automated Validation of Internet Security Protocols and Applications) tools and confirm its security attributes.

Keywords: Authenticated Key Agreement; Elliptic Curve; Impersonation Attack; Password Guessing Attack; Session Initiation Protocol; AVISPA tools.

1. Introduction

The session initiation protocol (SIP) is an application layer signalling protocol for creating, modifying, and terminating multimedia sessions among one or more participants. SIP was developed by the Internet Engineering Task Force (IETF) in 1996. With the widespread application of the Voice over IP (VoIP) in Internet [1] and mobility management [2{4], SIP has been receiving a lot of attention and the security of SIP is becoming increasingly important [5]. When a user wants to access a SIP service, he or she has to perform an authentication process from the remote server. Thus, authentication is one of the most important issues for SIP. Various authentication schemes, especially based on Elliptic Curve Cryptography (ECC), have been proposed to provide security for SIP for a decade [6–12].

In 2005, Yang et al. [13] indicated that the original SIP authentication scheme is vulnerable to off-line password guessing attack and server-spoofing attack. To overcome the attacks, Yang et al. proposed a modified scheme based on Diffie-Hellman key exchange protocol. However, Huang et al. [14] pointed out that the Yang et al.'s scheme may not be suitable for users with limited computational power and further proposed a new scheme. In [15], Jo et al. demonstrated that the schemes by Yang et al. and Huang et al. are both vulnerable to off-line password guessing attack.

Based on Yang et al.'s scheme, Durlanik and Sogukpinar [16] introduced an efficient authentication scheme for SIP by using Elliptic Curve Diffie-Hellman (ECDH) key exchange protocol. Because of the adoption of elliptic curves, Durlanik and Sogukpinar's scheme reduced the total execution time and the requirements for memory in comparison with Yang et al.'s scheme. However, Yoon and Yoo [17] indicated that Durlanik and Sogukpinar's scheme still suffered from off-line password guessing and Denning-Sacco attacks, and projected an improved scheme to overcome the weaknesses. However, Liu and Koenig [18] demonstrated that Yoon and Yoo's scheme still puts up with off-line password guessing and insider attacks.

In 2009, Tsai [19] proposed an efficient authentication protocol based on random nonce, in which one-way hash functions and exclusive-or operations were only utilized for computing all the communication messages. As a result, the computation cost was very low and it was suitable for low computation equipment. However, it was still

defenceless to off-line password guessing, Denning-Sacco and stolen-verifier attacks, furthermore, it did not provide any key agreement, known-key secrecy and perfect forward secrecy (PFS) [20–22]. To deal with the problems, Arshad and Ikram proposed an ECC-based authentication scheme [22]. But, Tang and Liu [23] demonstrated the vulnerability of Arshad and Ikram's scheme to off-line password guessing attack and introduced an improved scheme to overcome the weakness.

In 2010, Yoon et al. [24] also proposed an authentication scheme based on ECC to deal with the problems in Tsai's scheme [19]. In 2012, Xie [25] pointed out that Yoon et al.'s scheme still suffers from stolen-verifier and off-line password guessing attacks, and proposed a new security enhanced scheme for SIP to solve these problems. However in this paper, we indicate that the Xie's scheme is still vulnerable to impersonation attack, by which an active adversary can easily forge the identity of the server. Based on this attack, we also show that the Xie's scheme still suffers from off-line password guessing attack. Then, we propose an improved scheme to enhance the security of the Xie's scheme. Our improved scheme does not only maintain the merits and cover the demerits of the Xie's scheme, but also meets all the requirements of such schemes. Our scheme also provides mutual authentication with key agreement. Moreover, our scheme provides a password change phase. Specifically, the users could renew their passwords anytime and anywhere. Finally, the security analysis is presented.

Typically, the theoretical analysis of cryptographic protocols is normally used to verify the security attributes in the design. However, it is not sufficient, and simulation tools must also be employed to verify all the security requirements of the protocol. AVISPA [26] is a strong simulation engine for automated security analysis of cryptographic protocols [32]. Therefore, we make use of the AVISPA tools to confirm the security attributes of the proposed protocol.

The rest of this paper is organized as follows. In Section 2, we review the Xie's authenticated key agreement for session initiation protocol. In Section 3, we propose impersonation attack and off-line password guessing attack on the Xie's scheme. An enhanced authentication scheme for SIP is proposed in Section 4. The proposed protocol is then analyzed for security by the use of theoretical analysis and AVISPA tools in Section 5. In Section 6, we make a comparison between our scheme and some related schemes. Finally, Section 7 concludes the paper.

2. A brief review of the Xie's scheme

This section briefly reviews the Xie's authentication scheme for SIP [25]. The Xie's scheme consists of three phases: the setup phase, the registration phase and the authentication phase. The notations used in this paper are shown in Table 1.

2.1. System setup phase

In this phase, the server S sets the following system parameters: let q be a large prime number, $E(GF_q)$ an elliptic curve group defined over a finite field GF_q, P a generator of $E(GF_q)$of order q, and $h(.)$ a cryptographic hash function. S also selects an integer $K_s \in (1; q)$ as the long-live secret key, and computes the corresponding public key $Q_s = K_s P$. At the end of this phase, S publishes all parameters except K_s.

2.2. Registration phase

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

R1: U sends password PW to S via a secure channel.

R2: S computes $V\ PW = EK_s(PW)$ and stores $V\ PW$ to the user account database (i.e., the registration table) corresponding to U's information.

2.3. Authentication phase

If the legal user U wants to login into S, U and S perform the following steps:

A1: $U \rightarrow S$: REQUEST$\{username, aP\}$
U chooses a random integer $a(\in 1, q)$, computes and sends aP together with his or her username in a request message to S.

A2: $S \rightarrow U$: CHALLENGE$\{realm, bP, \sigma, \mathrm{KP_x}\}$
Upon receiving the request message, S rst randomly chooses $b, k \in (1, q)$ and computes bP, $SK_s = baP$, $\left(k \cdot h(SK_s \| bP)\right)P = (KP_x, KP_y)$ and $\sigma = k - \left(h(SK_s \parallel bP)\right)^{-1} K_s (\mathrm{mod}\ q)$. Then, S sends the challenge message CHALLENGE$\{realm, bp, \sigma, kP_x\}$ back to U.

A3: $U \rightarrow S$: RESPONSE $\{username, realm, h(username \| realm \| SK_u \| PW)\}$
Upon receiving the challenge message, U computes $SK_u = abP$, $r = \sigma \cdot h(SK_u \parallel bP)P + Q_s = (r_x; r_y)$ and checks if $r_x = KP_x$. If so, U computes $h(username \parallel realm \parallel SK_u \parallel PW)$ and sends RESPONSE$\{username, realm, h(username \| realm \| SK_u \| PW)\}$ back to S. Otherwise, U rejects it.

A4: Upon receiving the response message, S computes $PW = DK_S(V\ PW)$ and $h(username \parallel realm \parallel SK_u \parallel PW)$, and verifies if $h(username \parallel realm \parallel SK_u \parallel PW) = h(username \parallel realm \parallel SK_u \parallel PW)$. If so, U is authenticated. Otherwise, S aborts the session.

At the end of the execution of the protocol, the session key shared between U and S is set to $SK = h(SK_u \| P) = h(SK_S \| P)$.

Table 1. The Notations

Notation	Description
U	a user
username	the identity of the user U
realm	client's realm is used to prompt the username and password.
PW	the password of the user U
$V\ PW$	the password verifier of the user U
S	a remote server
K_s	the long-live secret key of the server
Q_s	the long-live public key of the server
SK	a session key
$h(.)$	a strong cryptographic one-way hash function
$E_{k_s}(.)$	a secure symmetric encryption algorithm under the secret key of the server
$D_{k_s}(.)$	a secure symmetric decryption algorithm under the secret key of the server
q	a large prime number
GF_q	a finite field with order q
$E(GF_q)$	an elliptic curve group defined over GF_q
P	a generator of $E(GF_q)$ with order q
$\|$	the string concatenation operation
$\oplus$	the exclusive-or operation

3. Cryptanalysis of the Xie's scheme

3.1. The proposed impersonation attack on the Xie's scheme

In this section, we show that the Xie's scheme is vulnerable to impersonation attack. We show that an active adversary can easily introduce himself to the users as a legal server. The proposed attack works as follows:

I1: When the legal user U wants to login into the server S, sends the request message $\{username, aP\}$ to the S.

I2: An active adversary $\mathcal{A}$ may eavesdrop the communication flows between U and S, intercept the request message $\{username, aP\}$, and do the following steps:

I2.1: Select a random number $b'\epsilon(1;\ q)$ and compute $b'P$ and $SK_s' = b'aP$.

I2.2: Select another random number $\sigma' \in (1, q)$ as a signature and compute

$$s'(h(SK_s'\|b'P\ ))P + Q_s = (KP_x', KP_y') \qquad (1)$$

I2.3: Send the challenge message $\{realm, b'P, \sigma', KP_x'\}$ back to the user U on behalf of S.

I3: Upon receiving the challenge message, U computes $SK_u = ab'P$ and

$$r = \sigma'\big(h(SK_u\|b'P)\big)P + Q_s = (r_x, r_y), \qquad (2)$$

and checks if $r_x = KP_x'$. The following statements indicate that the verification equation 2 holds:

$$\begin{aligned}(r_x, r_y) &= s'\big(h(SK_u\|b'P)\big)P + Q_s\\ &= s'\big(h(b'P\|b'P)\big)P + Q_s\\ &= s'\big(h(SK_s'\|b'P)\big)P + Q_s\\ &= (KP_x', KP_y').\end{aligned}$$

Thus, U would believe that the received mes-sage is generated by the legal server S. Then, U computes $h(username \parallel realm \parallel SK_u \parallel PW)$ and sends the response message $\{username, realm, SK_u, PW)$ to S. Finally, U computes the session key $SK = h(SK_u\|P)$.

I4: The adversary intercepts the response message and computes the session key $SK = h(SK_s'\|P)$.

As can be clearly seen, the session key shared between U and the adversary $\mathcal{A}$ is set to $SK = h(SK_u\|P) = h(SK_s'\|P) = h(ab'P\|P)$. Thus, the adversary without knowing the password PW and the server's private key K_s can easily impersonate the server S and share a secret key with U.

3.2. The proposed off-line password guessing attack on the Xie's scheme

As a result of the impersonation attack (see Section 3.1), off-line password guessing attack also can be applied to the Xie's scheme by an active adversary. To do so, the adversary $\mathcal{A}$ applies an impersonation attack and obstructs the response message $\{username, realm, h(username \parallel realm \parallel SK_u\|PW)\}$ at the end of the Step I3 in Section 3. After receiving the response message, $\mathcal{A}$ launches the off-line password guessing attack as follows:

G1: A selects a candidate password PW' from the uniformly distributed dictionary of size $|D|$.

G2: As mentioned in the impersonation attack (see Section 3.1, Step I2.1), the session key $SK_u = SK_s' = ab'P$, is known for the adversary. Therefore, $\mathcal{A}$ can compute $h(username \parallel realm \parallel SK_s' \parallel PW')$.

G3: $\mathcal{A}$ compares $h(username \parallel realm \parallel SK_s' \parallel PW')$ with $h(username \parallel realm \parallel SK_u \parallel PW)$. If they are equal, $\mathcal{A}$ guesses the right password of U. Otherwise, the adversary goes to the step G1 and does the next loop.

4. The proposed scheme for SIP

This section proposes an enhanced authentication scheme for session initiation protocol in order to overcome the above mentioned problems with the Xie's scheme. The proposed protocol contains four phases: system setup phase, registration phase, login and authentication phase, and password change phase.

4.1. System setup phase

In the system setup phase, S generates the following system parameters: an elliptic curve E over a finite field GF_q, an additive group of points on the elliptic curve $E(GF_q)$, the generating point P on $E(GF_q)$ of order q and a secure hash function $h(.)$. S also selects an integer $K_s \in (1, q)$ as the long-live secret key, and computes $Q_s = K_s P$ as the corresponding public key. Finally, S publishes the parameters $\{E(GF_q), P, q, h(.), Q_s\}$.

4.2. Registration phase

Figure 1 shows the registration phase of our scheme. When a user wants to login into the remote server, he/she firstly should register to the remote server. In this phase, the user communicates with the server through a secure channel. The details of this phase are as follows.

R1: The user freely chooses his or her $username$ and password PW, and interactively sends them to the server through a secure channel.

R2: The server computes $V\ PW = h(username \parallel K_s) \oplus h(username \parallel PW$ and stores $(username, V\ PW)$ in its database.

4.3. Login and authentication phase

Figure 1 shows the login and authentication phase of our scheme. In this phase, the user communicates with the remote server through a public channel. When the user U wants to login into the remote server, he or she performs the following steps to execute a session of the protocol:

A1: $U \rightarrow S$: REQUEST$\{username, aP\}$

U chooses a random integer $a \in (1, q)$, computes and sends aP in the request message REQUEST$\{username, aP\}$ to S.

A2: $S \rightarrow U$: CHALLENGE$\{realm, bP, \sigma\}$

Upon receiving the request message, S first randomly chooses $b \in (1, q)$ and computes bP, $SK_s = baP$ and $\sigma = h(SK_s \parallel K_s aP \parallel bP \parallel aP)$. Then, S sends the challenge message CHALLENGE$\{realm,\ bP, \sigma\}$ back to U.

A3: $U \rightarrow S$: RESPONSE$\{realm, H\}$

Upon receiving the challenge message, U computes $SK_u = abP$ and checks if $\sigma = h(SK_u \parallel aQ_s \parallel bP \parallel aP)$. If so, U computes $H = h(realm \parallel SK_u \parallel h(username \parallel PW)$ and sends RESPONSE$\{realm, h(realm \parallel H)\}$ back to S and computes the session key $SK = h(username \parallel SK_u \parallel aP \parallel bP)$. Other-wise, U rejects it.

A4: Upon receiving the response message, S verifies if $h(realm \parallel SK_s \parallel \{V\ PW \oplus h(username \parallel K_s)\}) = H$.

If so, U is authenticated and S computes the session key $SK = h(username \parallel SK_s \parallel aP \parallel bP)$. Otherwise, S aborts.

Finally, the session key shared between U and S is set to

$$SK = h(username \parallel SK_u \parallel aP \parallel bP)$$
$$= h(username \parallel SK_s \parallel aP \parallel bP)$$

4.4. Password change phase

Figure 1 shows the password change phase of our scheme. The user U can change the password freely in this phase. To do so, he/she firstly needs to execute the login and authentication phase with his/her username and old password PW. After receiving the successful authentication confirmation from the server and sharing the session key SK, the user U inputs the new password PW^* as follows:

C1. $U \rightarrow S$: $\{PWD, V\}$

The user U computes $PWD = h(SK \parallel username) \oplus h(username \parallel PW^*)$ and $V = h(SK \parallel h(username \parallel PW^*))$, an sends them to the server.

C2. $S \rightarrow U$: $\{Accept, R_1\}$ or $\{Reject, R_2\}$

Upon receiving the message PWD and V, the server computes $H_2' = PWD \oplus h(SK \parallel username)$ and checks whether V is equal to $h(SK \parallel H_2')$. If so, the server accepts the password change request, computes $R_1 = h(Accept \parallel username \parallel PWD \parallel V \parallel SK)$ and sends $\{Accept, R_1\}$ back to the user. Otherwise, they are not equal, the server rejects the password change request, computes $R_2 = h(Reject \parallel username \parallel PWD \parallel V \parallel SK)$ and sends $\{Reject, R_2\}$ back to the user. Finally, the server replaces $V\ PW$ with $VPW^* = h(username \parallel K_s) \oplus H_2'$.

It is obvious that the verification equation $h(realm \parallel SK_s \parallel \{VPW^* \oplus h(username \parallel K_s)\}) = H$ in Section 4.3, item A.4 is passed because

$$VPW^* = h(username \parallel K_s) \oplus H_2'$$
$$= h(username \parallel K_s) \oplus PWD \oplus h(SK \parallel username)$$
$$= h(username \parallel K_s) \oplus h(SK \parallel username)$$
$$= h(username \parallel PW^*) \oplus h(SK \parallel username)$$
$$= h(username \parallel K_s) \oplus h(username \parallel PW^*)$$

and,

$$h(realm \parallel SK_s \parallel \{VPW^* \oplus h(username \parallel K_s)\})$$
$$= h(realm \parallel SK_s \parallel h(username \parallel PW^*))$$
$$= h(realm \parallel SK_u \parallel h(username \parallel PW^*))$$
$$= H$$

User	Server
Registration phase	
select $username$ and PW	
	$\overset{\{username, PW\}}{\Longrightarrow}$
	compute $VPW = h(username\|\|K_s) \oplus h(username\|\|PW)$ store $\{username, PW\}$
Login and authentication phase	
randomly choose $a \in (1, q)$ compute aP	
	$\xrightarrow{\{username, aP\}}$
	randomly choose $b \in (1, q)$ compute bP, $SK_s = baP$ and $\sigma = h(SK_s \Vert K_s aP \Vert bP \Vert aP)$
	$\xleftarrow{\{realm, bP, \sigma\}}$
compute $SK_u = abP$ and check if $\sigma = h(SK_u \Vert aQ_s \Vert bP \Vert aP)$ If so, compute $H = h(realm \Vert SK_u \Vert h(username \Vert PW))$	
	$\xrightarrow{\{realm, H\}}$
and the session key $SK = h(username \Vert SK_u \Vert aP \Vert bP)$	verify if $h(realm \Vert SK_s \Vert \{VPW \oplus h(username \Vert K_s)\}) = H$ If so, compute the session key $SK = h(username \Vert SK_s \Vert aP \Vert bP)$
Password change phase	
randomly select a new password PW^* compute $H_2 = h(username \Vert PW^*)$ $PWD = h(SK \Vert username) \oplus H_2$ and $V = h(SK \Vert h(username \Vert PW^*))$	
	$\xrightarrow{\{PWD, V\}}$
	compute $H_2' = PWD \oplus h(SK \Vert username)$ check if $V = h(SK \Vert H_2')$ if so, accept the password change request, compute $R_1 = h(Accept \Vert username \Vert PWD \Vert V \Vert SK)$ compute $VPW^* = h(username \Vert K_s) \oplus H_2'$ replace VPW with VPW^* otherwise, reject the password change request, compute $R_2 = h(Reject \Vert username \Vert PWD \Vert V \Vert SK)$
	$\xleftarrow{\{Accept/Reject, R_1/R_2\}}$

Figure 1. The proposed protocol

5. Security analysis

5.1. Theoretical analysis

Replay attack. Suppose an attacker $\mathcal{A}$ intercepts REQUEST$(username, aP)$ from U in step A1 and replays it to impersonate U. However, $\mathcal{A}$ cannot compute a correct session key $SK = abP$ and deliver it to S in step A3 unless he/she can correctly guess the password PW and guess b from bP or a from aP. When $\mathcal{A}$ tries to guess a from aP or b from bP, he/she will face the Elliptic Curve Discrete Logarithm Problem (ECDLP) which is untraceable. On the other hand, suppose $\mathcal{A}$ intercepts CHALLENGE$(realm, bP, s)$ from S in step A2 and replays it to impersonate S. The replied message cannot pass the verification process $\sigma = h(SK_u \parallel aQ_s \parallel bP \parallel aP)$, since a is a new nonce chosen by U in each session and the adversary has no control of it. Therefore, the proposed scheme can resist the replay attack.

Stolen-verifier attack. When attacker $\mathcal{A}$ steals verifier $VPW = h(username \parallel K_s) \oplus h(username \parallel PW)$ from the database of the server, he/she cannot obtain the right password PW from VPW without knowing the secret key K_s of the server, which is a high entropy number and cannot be guessed by enumeration. Therefore, the proposed scheme is secure against stolen-verifier attack.

Denning-Sacco attack. Attacker $\mathcal{A}$ may obtain the session key $SK = h(username \parallel SK_u \parallel aP \parallel bP) = h(username \parallel SK_s \parallel aP \parallel bP)$ for some reasons, but he/she cannot obtain user's secret password PW and server's secret key K_s because he/she will face to obtain abP which is protected by a hash function.

Impersonation attack. An adversary $\mathcal{A}$ cannot masquerade as server, because he/she cannot compute the signature $\sigma = h(SK_s \parallel K_s aP \parallel bP \parallel aP)$ with-out knowing the server's secret key K_s. $\mathcal{A}$ also cannot impersonate the user to authenticate with the server, because he/she cannot construct the message $RESPONSE\{username, realm, h(realm \parallel SK_u \parallel h(username \parallel PW)\}$ without the knowledge of PW. Therefore, the proposed scheme resists im-personation attack.

Password guessing attack. It is divided into online password guessing attack and off-line password guessing attack. Online password guessing attack can be preserved by limiting the login times. The exchanged messages between the user and the server, in the login phase (step A1) and verification phase (step A2), are independent of the user's password; therefore the adversary cannot execute any off-line guessing attack on our scheme.

Man-in-the-middle attack. Password PW of U and the secret key K_s of S are used to prevent the man-in-middle attack. Therefore, the active adversary $\mathcal{A}$ cannot intrude into the communication between S and U to intercept the exchanged data and inject false information.

Modification attack. An adversary $\mathcal{A}$ cannot modify the communicated messages $(username, aP)$ in step A1, $(realm, bP, \sigma)$ in step A2 and $\{username, realm, h(realm \parallel SK_u \parallel h(username \parallel PW))\}$ in step A3, because the user and the server detect them by verifying s and $h(realm \parallel SK_u \parallel h(username \parallel PW))$, respectively.

Known-key security. In this attack, an adversary, who has some previous session keys, is willing to compute the next session keys. Assume that some previous session keys are known for the adversary $\mathcal{A}$. It does not give any useful information to $\mathcal{A}$ for computing the next session keys, because the short-term private keys a and b are changed in each session. Note that, A cannot obtain a from aP or b from bP because he/she will face the ECDLP. Therefore, the proposed protocol satisfies the known-key security.

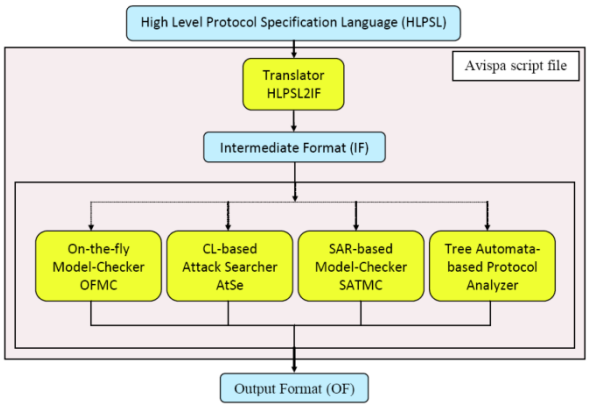

Figure 2. The architecture of the AVISPA tools

Perfect forward secrecy. Perfect forward secrecy means that if long-term private keys of one or more entities are compromised, the secrecy of previous session keys established by the trusted entities is not affected. In the proposed protocol, the adversary who knows PW and K_s cannot determine the previous session keys because long-term private keys are not utilized for computing the session keys. In addition, the adversary cannot compute neither a nor b from s, aP, bP and K_s since he/she has to solve Elliptic Curve Deffie-Hellman Problem (ECDHP). Therefore, the proposed protocol satis es the perfect forward secrecy.

5.2. Simulation results

In the last decade, we have witnessed the development of a large number of new techniques for the formal analysis of security protocols. Until now, many (semi-)automated security protocol analysis tools have been proposed (e.g., [26-28]). One of the tools that has seen the widest use is the AVISPA [26] which is a push-button tool for the automated

validation of Internet security-sensitive protocols and applications. It provides a modular and expressive formal language for specifying protocols and their security properties, and integrates different back-ends that implement a variety of state-of-the-art automatic analysis techniques.

The architecture of AVISPA is shown in Figure 2. The firrst step in using the tool is to present the analyzed protocol in a special language called High Level Protocol Specification Language (HLPSL). The HLPSL presentation of the protocol is translated into the lower level language called Intermediate Format (IF). This translation is performed by the translator called HLPSL2IF. This step is totally transparent to the user. IF presentation of the protocol is used as an input to the four different back-ends: On-the-fly Model-Checker (OFMC), CL-based Attack Searcher (CL-AtSe), SAT-based Model-Checker (SATMC) and Tree-Automata-based Protocol Analyzer (TA4SP). These back-ends perform the analysis and output the results in precisely defined output format stating whether there are problems in the protocol or not.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/avispa /web-interface-computation/.
  /tempdir/workfileaTPT9E.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime    : 0.00 s
  searchTime   : 0.09 s
  visitedNodes : 21 nodes
  depth        : 4 plies
```

Figure 3. The output of OFMC back-end

In order to evaluate the security of the proposed protocol by the AVISPA tools the protocol is coded in HLPSL. The HLPSL code of the proposed protocol is included in Appendix A. After execution of the code in AVISPA tool, the outputs of OFMC (Figure 3), CL−AtSe (Figure 4) and SATMC (Figure 5) back-ends were generated. According to the summary results, the proposed protocol is SAFE and there are no major attacks on it. Therefore, these results confirm the theoretical analysis in Section 5.1.

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
  /home/avispa /web-interface-computation /.
  /tempdir/workfileaTPT9E.if
GOAL
  As Specified
BACKEND
  CL_AtSe
STATISTICS
  Analysed     : 26 states
  Reachable    : 15 states
  Translation  : 0.01 seconds
  Computation  : 0.00 seconds
```

Figure 4. The output of CL-AtSe back-end

```
SUMMARY
  SAFE
DETAILS
  STRONGLY_TYPED_MODEL
  BOUNDED_NUMBER_OF_SESSIONS
  BOUNDED_SEARCH_DEPTH
  BOUNDED_MESSAGE_DEPTH
PROTOCOL
  /home/avispa /web-interface-computation/.
  /tempdir/workfileaTPT9E.if
GOAL
  %% see the HLPSL specification. .
BACKEND
  SATMC
COMMENTS
STATISTICS
  attackFound          false     boolean
  upperBoundReached    true      boolean
  graphLeveledOff      3         steps
  satSolver            zchaff    solver
  maxStepsNumber       11        steps
  stepsNumber          3         steps
  atomsNumber          0         atoms
  clausesNumber        0         clauses
  encodingTime         0.06      seconds
  solvingTime          0         seconds
  if2sateCompilationTime 1.84    seconds
ATTACK TRACE
  %% no attacks have been found. .
```

Figure 5. The output of SATMC back-end

6. Security and performance comparison

In this section, we evaluate the performance and functionality of our proposed protocol and make comparisons with some related authenticated key agreement for session initiation protocols. Table 2 shows the main computation cost of our scheme. Table 3 shows the performance comparisons of our proposed protocol and some other related protocols.

We mainly consider the computations of login and authentication phase and session key agreement since these are the principal parts of an authentication protocol and should be implemented for each session. In Table 3, it is obvious that the computation cost of the proposed protocol is lesser than the Xie's scheme. However, it is worth several additional hash operations to achieve the security and functionality properties.

Table 4 lists the security comparisons among our pro-posed protocol and other related protocols. It demonstrates that our protocol has many excellent features and is more secure than other related protocols.

Table 2. Computation cost of login and authentication phase

	User	Server	Total
No. of scale multiplication	3	3	6
No. of hash function	4	4	8
No. of exclusive or	0	1	1

7. Conclusions

In this paper, we briefly reviewed the Xie's authenticated key agreement protocol session initiation protocol. We demonstrated that the Xie's scheme is vulnerable to the impersonation attack in which an active adversary with-out knowing the users' password and the server's private key can easily impersonate the server to the users and share secret keys with them. As a result of the impersonation attack, we pointed out that the Xie's scheme also suffers from the off-line password guessing attack. The main aw of the Xie's scheme is due to the signature scheme used by the server which is forgeable. To overcome the security weaknesses, we proposed an improved scheme. In comparison to the related schemes, the proposed scheme not only is secure against well-known crypto-graphical attacks such as guessing attacks, replay attacks, but also provides mutual authentication, perfect forward secrecy and secure password change.

Table 3. Comparison of computation costs

	Durlanik [16]	Yang [13]	Tsai [19]	Yoon [24]	Arshad [22]	Tang [23]	Xie [25]	Ours
No. of exponentiation	0	4	0	0	0	0	0	0
No. of scale multiplication	4	0	0	6	5	4	6	6
No. of point addition	0	0	0	3	0	2	1	0
No. of hash-to-point	0	0	0	0	0	2	0	0
No. of hash function	6	8	7	4	8	7	6	8
No. of exclusive or	4	4	3	0	2	1	0	1
No. of modular inverse	0	0	0	0	0	0	1	0
No. of symmetric key encryption	0	0	0	0	0	0	2	0
Security	ECDLP	DLP	HASH	ECDLP	ECDLP	ECDLP	ECDLP	ECDLP

Table 4. Comparison of security attributes

	Durlanik [16]	Yang [13]	Tsai [19]	Yoon [24]	Arshad [22]	Tang [23]	Xie [25]	Ours
Reply attack	Secure	Secure	Secure	Secure	Secure	Secure	Secure	Secure
Man-in-the-middle attack	Secure	Secure	Insecure	Secure	Secure	Secure	Insecure	Secure
Impersonation attack	Insecure	Insecure	Insecure	Secure	Insecure	Secure	Insecure	Secure
Password guessing attack	Insecure	Insecure	Insecure	Insecure	Insecure	Secure	Insecure	Secure
Denning-Sacco attack	Insecure	N/A	Insecure	Insecure	Secure	Secure	Secure	Secure
Stolen-verifier attack	Insecure	Insecure	Insecure	Insecure	Secure	Secure	Secure	Secure
Mutual authentication	Provided	Provided	Provided	Provided	Provided	Provided	Provided	Provided
Session key security	Provided	N/A	Provided	Provided	Provided	Provided	Provided	Provided
Known key secrecy	Provided	N/A	Not provided	Provided	Provided	Provided	Provided	Provided
Perfect forward secrecy	Provided	N/A	Not provided	Provided	Provided	Provided	Provided	Provided

N/A: Not Applicable or Not Available

References

[1] **W. E. Chen, Y. L. Huang, Y. B. Lin**. An effective IPv4-IPv6 translation mechanism for SIP applications in next generation networks. *International Journal of Communication Systems*, 2010, Vol. 23, No. 8, 919–928.

[2] **C. C. Lee, I. E. Liao, M. S. Hwang**. An ex-tended certificate-based authentication and security protocol for mobile networks. *Information Technology and Control*, 2009, Vol. 38, No. 1, 61-66.

[3] **C. T. Li**. A more secure and efficient authentication scheme with roaming service and user anonymity for mobile communications. *Information Technology and Control*, 2012, Vol. 41, No. 1, 69-76

[4] **Y. H. Cheng, F. M. Chang, S. J. Kao**. Efficient hierarchical SIP mobility management for WiMAX networks. *Computers & Mathematics with Applications*, 2012, Vol. 64, No. 5, 1522-1531.

[5] **D. Geneiatakis, T. Dagiuklas, G. Kambourakis, C. Lambrinoudakis, S. Gritzalis, S. Ehlert, D. Sisalem**. Survey of security vulnerabilities in session initiation protocol. *IEEE Communications Surveys and Tutorials*, 2006, Vol. 8, No. 3, 68-81.

[6] **S. S. Mousavi-Nik, M. H. Yaghmaee-Moghaddam, M. B. Ghaznavi-Ghoushchi**. Proposed secure SIP authentication scheme based on elliptic curve cryptography. *International Journal of Computer Applications*. 2012, Vol. 58, No. 8, 25-30.

[7] **E. J. Yoon, K. Y. Yoo, C. Kim, Y. S. Hong, M. Jo, H. H. Chen**. A secure and efficient SIP authentication scheme for converged VoIP networks. *Computer Communications*, 2010, Vol. 33, No. 14, 1674-1681.

[8] **L. Wu, Y. Zhang, F. Wang**. A new provably secure authentication and key agreement protocol for SIP using ECC. *Computer Standards & Interfaces*, 2009, Vol. 31, No. 2, 286-291.

[9] **Y.P. Liao, S. S. Wang**. A new secure password authenticated key agreement scheme for SIP using self-certified public keys on elliptic curves. *Computer Communications*, 2010, Vol. 33, No. 3, 372-380.

[10] **S. Wu, Q. Pu, F. Kang**. Practical authentication scheme for SIP. *Peer-to-Peer Networking and Applications*. 2013, Vol. 6, No. 1, 61-74.

[11] **D. He, J. Chen, Y. Chen**. A secure mutual authentication scheme for session initiation protocol using elliptic curve cryptography. *Security and Communication Networks*, 2012, Vol. 5, No. 12, 1423–1429.

[12] **J. W. Hong, S. Y. Yoon, D. I. Park, M. J. Choi, E. J. Yoon, K. Y. Yoo**. An new efficient key agreement scheme for VSAT satellite communications based on elliptic curve cryptosystem. *Information Technology and Control*, 2011, Vol. 40, No. 3, 252–259.

[13] **C. C. Yang, R. C. Wang, W. T. Liu**. Secure authentication scheme for session initiation protocol. *Computers & Security*, 2005, Vol. 24, No. 5, 381-386.

[14] **H. F. Huang, W.C. Wei, G. E. Brown**. A new efficient authentication scheme for session initiation protocol. In: *9th Joint Conference on Information Sciences*, 2006.

[15] **H. Jo, Y. Lee, M. Kim, S. Kim, D. Won**. Off-line password-guessing attack to Yang's and Huang's authentication schemes for session initiation protocol. *Fifth International Joint Conference on INC, IMS and IDC*, 2009, pp. 618-621.

[16] **A. Durlanik, I. Sogukpinar**. SIP authentication scheme using ECDH. *World Enformatika Socity Transations on Engineering Computing and Technology*, 2005, Vol. 8, 350-353.

[17] **E. J. Yoon, K. Y. Yoo**. Cryptanalysis of DS-SIP authentication scheme using ECDH. In: *International Conference on New Trends in Information and Service Science*, 2009, pp. 642-647.

[18] **F. W. Liu, H. Koenig**. Cryptanalysis of a SIP authentication scheme. In: *12th IFIP TC6/TC11 International Conference, CMS 2011, Lecture Notes in Computer Science*, 2011, Vol. 7025, 134-143.

[19] **J. L. Tsai**. Efficient nonce-based authentication scheme for session initiation protocol. *International Journal of Network Security*, 2009, Vol. 8, No. 3, 312–316.

[20] **E. J. Yoon, K. Y. Yoo**. A new authentication scheme for session initiation protocol. In: *International Conference on Complex, Intelligent and Soft-ware Intensive Systems, CISIS'09*, 2009, pp. 549-554.

[21] **T. H. Chen, H. L. Yeh, P. C. Liu, H. C. Hsiang, W. K. Shih**. A secured authentication protocol for SIP using elliptic curves cryptography. In: *FGCN 2010, Part I, Communications in Computer and Information Science*, 2010, Vol. 119, pp. 46-55.

[22] **R. Arshad, N. Ikram**. Elliptic curve cryptography based mutual authentication scheme for session initiation protocol. *Multimedia Tools and Applications*, 2013, Vol. 66, No. 2, 165-178.

[23] **H. Tang, X. Liu**. Cryptanalysis of Arshad et al.'s ECC-based mutual authentication scheme for session initiation protocol, *Multimedia Tools and Applications*, 2013, Vol. 65, No. 3, 165-178.

[24] **E. J. Yoon, Y. N. Shin, I. S. Jeon, K. Y. Yoo**. Robust mutual authentication with a key agreement scheme for the session initiation protocol. *IETE Technical Review*, 2010, Vol. 27, No. 3, 203-213.

[25] **Q. Xie**. A new authenticated key agreement for session initiation protocol. *International Journal of Communication Systems*, 2012, Vol. 25, No. 1, 47-54.

[26] The AVISPA Project. http://www.avispa-project.org.

[27] **B. Blanchet**. An effcient cryptographic protocol verifier based on Prolog rules. In: *14th IEEE Computer Security Foundations Workshop (CSFW)*, 2001, pp. 82-96.

[28] **C. Cremers**. The Scyther Tool: verification, falsification, and analysis of security protocols. In: *CAV08, LNCS*, 2008, Vol. 5123, pp. 414-418.

Received September 2012.

Appendix A. HLPSL code of the proposed protocol

```
role client(
    A, S        : agent ,
    SND,RCV     : channel (dy) ,
    H           : hash_func ,
    P, Qs       : public_key )

played by A
def=
    local State             : nat ,
    PW                      : symmetric_key,
    Kas, Rs, Ra, Sigma, SKu, Ta, Ts,F   : text,
    Username , Realm        : message
const sec_kas1 , sec_sku,
    sec_ra , sec pw         : protocol_lid

init State :=0
transition
1. State =0/\RCV(start)=|>
    State':=1
    / \ Ra':=new ()
    / \ SND({Ra'}_P . Username)
    / \ witness (A, S , na , Ra')
    / \ secret(Ra', sec_ra ,A)

2. State=1
    / \ RCV( Realm.{Rs'}_ P.H({Ra.Rs'}_P.
        {Ra'}_Qs.{Rs'}_P, {Ra'}_P)) = | >
    State':=2
    / \ F':=H(Realm.{Ra.Rs}_P.H(Username.PW))
    / \ SND(Realm.F')
    / \Kas':=H(Username.{Ra.Rs}_P.{Ra}_P.{Rs}_P)
    / \ secret(PW,sec_pw,A)
    / \ secret(Kas', sec_kas1, {A,S})
    / \ request (A, S, ns, Ra)
    / \ request (A, S, ns, PW)
end role
```

```
role server (
    S ,A                    : agent,
    SND,RCV                 : channel(dy),
    H                       : hash_func,
    P, Qs                   : public_key)

played_by S
def=
    local State             : nat,
      PW                    : symmetric_key,
      Ra, Rs, Sigma, SKs, Ta, Ts, F, Kas: text,
      Username, Realm       : message
const sec_kas2, sec_rs,  sec_pw1  : protocolid

init State :=0

transition

1. State =0/ \RCV({Ra'}_P.Username)=| >
    State':=1
    / \Rs':=new( )
    / \ Sigma':=H({Rs'.Ra}_P.{{Ra}_P}_
            inv (Qs ).{Rs'}_P,{Ra}_P)
    / \SND(Realm.{Rs'}_P.Sigma')
    / \ witness (S, A, ns, Rs ')
    / \ secret (Rs', sec_rs, S)
    / \ secret (PW, sec_pw1, S)
2. State=2 / \RCV(Realm.F')=| >
    State':=3
    / \F':=H(Realm.{Rs.Ra}_P.H(Username.PW))
    / \Kas':=H(Username.{Rs.Ra}_P.{Ra}_P.{Rs}_P)
    / \ secret (Kas', sec_kas2, {A, S})
    / \ request (S ,A, na , Rs)
end role
```

```
role session (
    A, S            : agent ,
    H               : hash_func  ,
    P, Qs           : public_key )

def=
local
SA, RA, SS, RS      : channel (dy)

composition

client (A, S, SA, RA, H, P, Qs)
/ \ server (S, A, SS, RS, H, P, Qs)
end role
```

```
role environment ( )
def=
const
na, ns              : protocolid ,
a, s, i             : agent ,
h                   : hash_func ,
p, qs, qi           : public_key

intruder_knowledge={a, s, h, p, qs, qi}

composition

session (a, s, h, p, qs)
/ \ session (a, i, h, p, qi)
/ \ session (i, s, h, qs, qi)
end role
```

```
goal
secrecy_of
    sec_kas1, sec_kas2, sec_ra, sec_rs, sec_pw,
    sec_pw1
authentication_on na
authentication_on ns
end goal
```

```
environment ( )
```