Xie and Tang *SpringerPlus (2016)5:1045*
DOI 10.1186/s40064-016-2725-0

SpringerPlus

RESEARCH Open Access

Biometrics based authentication scheme for session initiation protocol

Qi Xie* and Zhixiong Tang

*Correspondence:
qixie68@126.com
Key Laboratory
of Cryptography
and Network Security,
Hangzhou Normal University,
Hangzhou 311121, China

Abstract

Many two-factor challenge-response based session initiation protocol (SIP) has been proposed, but most of them are vulnerable to smart card stolen attacks and password guessing attacks. In this paper, we propose a novel three-factor SIP authentication scheme using biometrics, password and smart card, and utilize the pi calculus-based formal verification tool ProVerif to prove that the proposed protocol achieves security and authentication. Furthermore, our protocol is highly efficient when compared to other related protocols.

Keywords: Authentication, Three-factor, Key agreement, Session initiation protocol

Background

The session initiation protocol (SIP) is an application layer controlling protocol for creation, modification and termination of Voice over Internet Protocol (VoIP) sessions with one or more participants. With the rapid growth of VoIP users, SIP is used in both the wireless and the wired networks widely. Originally, SIP authentication scheme is derived from HTTP digest authentication (Franks et al. 1999), which cannot resist server-spoofing attack and password guessing attack (Yang et al. 2005). Since then, various user authentication schemes for SIP have been proposed.

In 2005, Yang et al. (2005) proposed a new SIP authentication scheme based on Diffie-Hellman key exchange protocol, but Huang and Wei (2006) found that Yang et al.'s scheme has high computational costs and proposed an efficient SIP scheme. To improve the efficiency, Durlanik and Sogukpinar (2005) and Wu et al. (2009) also proposed SIP authentication protocols using the Elliptic Curve Cryptography (ECC), respectively. Unfortunately, Yang et al.'s and Huang et al.'s schemes suffer from the off-line password guessing attack (Jo et al. 2009), while Durlanik et al.'s and Wu et al.'s schemes are vulnerable to the Denning-Sacco attack and the off-line password guessing attack (Yoon et al. 2010b). Yoon et al. (2010b) presented an improved scheme to overcome these weaknesses. But Liu and Koenig pointed out that Yoon et al.'s SIP authentication scheme is still insecure against the off-line password guessing attack and the insider attack (Liu and Koenig 2011). Applying one-way hash function and the fast logic operations like exclusive-or, Tsai (2009) proposed a nonce based SIP authentication scheme. Later on, Yoon et al. (2010a) demonstrated that their scheme is vulnerable to Denning-Sacco

attack, off-line password guessing attack and stolen-verifier attack, and proposed a new SIP authentication scheme. In 2012, Xie (2012) demonstrated that Yoon et al.'s scheme is still vulnerable to stolen-verifier attack and off-line password guessing attack, and proposed an improvement of Yoon et al.'s scheme, but Farash and Attari (2013) found that Xie's protocol is also insecure against impersonation attack and off-line password guessing attack, and then they proposed an improved scheme to resolve these problems.

Recently, to enhance the performance and secrecy, Arshad and Ikram (2013) proposed an ECC-based SIP authentication protocol in 2013. But Tang and Liu (2013), He et al. (2012) and Pu et al. (2013) pointed out that Arshad et al.'s protocol is vulnerable to off-line password guessing attack. They also developed new schemes to enhance the security of Arshad et al.'s scheme. Later, Irshad et al. (2014) demonstrated that Tang et al.'s scheme cannot resist the server impersonation attack if an adversary can obtain the user's password, and they proposed an improved protocol using ECC. Recently, Zhang et al. (2014) proposed a new password-based SIP authentication protocol, but Tu et al. (2015), Irshad et al. (2015) and Wu et al. (2013) showed that Zhang et al.'s protocol is vulnerable to the impersonation attack, and they proposed improved protocols respectively. After that, Arshad and Nikooghadam (2016) showed that Irshad et al.'s scheme is still vulnerable to impersonation attack. Farash (2016) and Mishra et al. (2016) found that Tu et al.'s protocol cannot resist the impersonation attack, and also presented improved schemes. It is worth mentioning that Mishra et al.'s scheme is a three-factor SIP authentication scheme, but it does not achieve perfect forward secrecy. Very recently, Chaudhry et al. (2015b) found that Tu et al.'s scheme is vulnerable to server impersonation attack. Moreover, both Tu et al.'s and Farash's improved schemes cannot protect user's privacy and suffer from replay and denial of services attacks. To enhance the security, they proposed a privacy preserving authentication scheme for SIP. Kumari et al. (2015) argued that Farash's protocol cannot withstand impersonation attack, password guessing attack, and session-specific temporary information attack. Further, Kumari et al. proposed an improved protocol to fix the weaknesses of Farash's protocol.

Many of above mentioned session initiation protocols are based on either password or both of password and smart card. However, password based protocol may suffer from password guessing attack, and smart card based protocol may suffer from smart card stolen attack by extracting information stored in smart card, even if the smart card is designed for achieving a certain level of tamper resistance (Witteman 2002). In order to solve password guessing attack and smart card stolen attack for SIP authentication scheme, we use user's biometrics to protect user's password and the sensitive information in smart card, since user's biometrics have many advantages, such as it is difficult to be fabricated, distributed, lost, forgotten, guessed or copied (Li and Hwang 2010). On the other hand, fuzzy extractor can always output the same random string if the input biometrics has sufficient similarity to the stored biometrics (Dodis et al. 2004). Therefore, in this paper, we propose a biometrics-based SIP authentication scheme, and use pi calculus (Abadi and Fournet 2001) based formal verification tool ProVerif (Abadi et al. 2009) to prove authentication and security of the proposed protocol.

The rest of the paper is organized as follows. In "Biometrics-based SIP authentication scheme" section, we propose our Biometrics-based SIP authentication scheme. Security analysis and formal verification are given in "Security analysis and formal verification"

section. "Security and performance comparisons" section compares the security and performance of our protocol to existing ones, and we conclude the paper in "Conclusions" section.

Biometrics-based SIP authentication scheme

A biometrics based SIP authentication scheme is proposed in this section, which consists of three phases: registration, login and authentication, and password change. In this section, we first describe the construction of the fuzzy extractor, then we give the scheme specification of the proposed biometrics based SIP.

Fuzzy extractor

Fuzzy extractor contains a pair of randomized procedures $\langle$"generate" (*Gen*), "reproduce" (*Rep*)$\rangle$. The procedure *Gen* is designed for inputting users' biometrics BIO, and then outputting a random and uniform string η as secret information as well as a random auxiliary string λ as public information, namely, $Gen(\text{BIO}) = (\eta, \lambda)$. The procedure *Rep* takes the biometrics BIO^* and the auxiliary string λ as inputs. Even if the inputted BIO^* has slightly difference with BIO, as long as the difference is less than the threshold, the procedure *Rep* will generate the same string η, namely, $Rep(BIO^*, \lambda) = \eta$. Though we cannot always get the same biometrics due to the impact of noisy data when sampling, fuzzy extractor can overcome this problem. Readers may refer to Dodis et al. (2004), Yang and Yang (2009) for the detailed introduction of fuzzy extractor. The notations used in this paper are given in Table 1.

Registration

A legal user U_i must register in the remote server S beforehand by performing the following steps, as shown in Algorithm 1.

Step 1. The user U_i chooses a password pw_i, a random number $a_i \in Z_n^*$, computes $M = h(a_i \| pw_i)$ and sends the register message $\{ID_i, M\}$ to S via a secure channel.

Step 2. After S receives the register request message $\{ID_i, M\}$, S computes $R = M \oplus h(ID_i \| x)$, stores R into a smart card and sends it to U_i through a secure channel.

Table 1 The notations

Notation	Description
E	An elliptic curve with large order n
P	A generator on E with large order n
U_i	The user U_i
BIO_i	The user U_i's biometrics
ID_i	The user U_i's identity
pw_i	The user U_i's password
S	The server S
x	The server S's secret key
$h()$	A secure one-way hash function
$\|$	A string concatenation operation
$\oplus$	A exclusive-or(XOR) operation

Step 3. After U_i obtains the smart card, he or she enters his or her biometrics BIO_i on a specific device and computes $Gen(BIO_i) = (\eta, \lambda)$, $B = a_i \oplus h(\eta)$, $C = h(ID_i \| pw_i \| a_i)$ and stores B, C and λ into the smart card. Thus, the smart card contains $\{B, C, \lambda, R\}$.

User(U_i)		Server(S)
Choose $a_i \in Z_n^*$, pw_i		
Compute $M = h(a_i \| pw_i)$		
	$\xrightarrow{\{ID_i, M\}}$	
		Compute $R = M \oplus h(ID_i \| x)$
		Store R into the smart card
	$\xleftarrow{\text{Smart card}}$	
Enter biometrics BIO_i		
Compute $Gen(BIO_i) = (\eta, \lambda)$		
$B = a_i \oplus h(\eta)$		
$C = h(ID_i \| pw_i \| a_i)$		
Store B, C and λ into the smart card		

Algorithm 1 Registration Phase

Login and authentication

In this phase, U_i and S can be authenticated by each other and establish the session key. The process is shown in Algorithm 2.

Step 1. The user U_i inserts his or her smart card into a card reader, inputs his or her identity ID_i and password pw_i, and enters biometrics BIO_i^*. The smart card selects a random number $b \in Z_n^*$, computes $Rep(BIO_i^*, \lambda) = \eta$, $a_i = B \oplus h(\eta)$, and $C' = h(ID_i \| pw_i \| a_i)$. Then, the smart card checks whether C' is equal to C. If they are not equal, the protocol is terminated; otherwise, compute $D = bP$, $D' = R \oplus h(a_i \| pw_i) = h(ID_i \| x)$ and $F = h(ID_i \| D \| D')$. At last, U_i sends the message $REQUEST\{ID_i, D, F\}$ to S.

Step 2. When the server S receives $REQUEST\{ID_i, D, F\}$, S computes $D'' = h(ID_i \| x)$ and checks if F and $h(ID_i \| D \| D'')$ are equal. If they are not equal, S rejects the request; otherwise, S randomly chooses two numbers $u, t \in Z_n^*$, computes $H = uP$, $K = u \cdot h(ID_i \| x)D$, $SK = h(ID_i \| t \| K)$ and $Auth_s = h(D \| K \| D'' \| t \| SK \| H)$. Finally, S sends the message $CHALLENGE\{realm, Auth_s, H, t\}$ to U_i.

Step 3. When the user U_i receives $CHALLENGE\{realm, Auth_s, H, t\}$, he or she computes $K = bD'H$ and $SK = h(ID_i \| t \| K)$. Then U_i checks if $Auth_s$ and $h(D \| K \| D' \| t \| SK \| H)$ are equal. U_i terminates the protocol if they are not equal; otherwise, U_i computes $Auth_u = h(ID_i \| realm \| K \| D' \| t \| SK \| H \| D)$ and sends the message $RESPONSE\{ID_i, realm, Auth_u\}$ to S.

Step 4. When the server S receives $RESPONSE\{ID_i, realm, Auth_u\}$, it checks whether $Auth_u$ is equal to $h(ID_i \| realm \| K \| D'' \| t \| SK \| H \| D)$. If so, S and U_i established the session key SK.

User(U_i)		Server(S)
Input ID_i, pw_i, BIO_i^*, select $b \in Z_n^*$		
Compute $Rep(BIO_i^*, \lambda) = \eta$		
$a_i = B \oplus h(\eta)$		
$C' = h(ID_i \| pw_i \| a_i)$		
Check $C' \stackrel{?}{=} C$		
Compute $D = bP$		
$D' = R \oplus h(a_i \| pw_i) = h(ID_i \| x)$		
$F = h(ID_i \| D \| D')$		
	$REQUEST\{ID_i, D, F\}$ →	
		Compute $D'' = h(ID_i \| x)$
		Check $F \stackrel{?}{=} h(ID_i \| D \| D'')$
		Choose $u, t \in Z_n^*$
		Compute $H = uP$
		$K = uh(ID_i \| x)D$
		$SK = h(ID_i \| t \| K)$
		$Auth_s = h(D \| K \| D'' \| t \| SK \| H)$
	← $CHALLENGE\{realm, Auth_s, H, t\}$	
Compute $K = bD'H$		
$SK = h(ID_i \| t \| K)$		
Check $Auth_s \stackrel{?}{=} h(D \| K \| D' \| t \| SK \| H)$		
$Auth_u = h(ID_i \| realm \| K \| D' \| t \| SK \| H \| D)$		
	$RESPONSE\{ID_i, realm, Auth_u\}$ →	
		Check $Auth_u \stackrel{?}{=} h(ID_i \| realm \| K \| D'' \| t \| SK \| H \| D)$

Algorithm 2 Login and Authentication Phase

Password change

The user U_i inserts his or her smart card into a terminal, inputs his ID_i, old password pw_i, new password pw_i^{new}, chooses a random number $a_i^{new} \in Z_n^*$ and enters biometrics BIO_i^* on a specific device. Then the smart card computes $Rep(BIO_i^*, \lambda) = \eta$, $a_i = B \oplus h(\eta)$. After this, the smart card verifies $h(ID_i \| pw_i \| a_i) = C$. If it does not hold, the smart card rejects the request; otherwise, the smart card computes $R^{new} = h(a_i^{new} \| pw_i^{new}) \oplus R \oplus h(a_i \| pw_i)$, $B^{new} = a_i^{new} \oplus h(\eta)$ and $C^{new} = h(ID_i \| pw_i^{new} \| a_i^{new})$, and replaces (R, B, C) with $(R^{new}, B^{new}, C^{new})$.

Security analysis and formal verification

In this section, we will analyze the security of the proposed scheme.

Formal verification

In order to prove the security of cryptographic protocols, there are some available formal verification tools, such as BAN logic (Burrows et al. 1989), AVISPA (Armando et al. 2005) and ProVerif. In this section, we prove secrecy and authentication using ProVerif, because it is performed automatically and efficiently, and can detect errors easily. ProVerif makes use of Dolev-Yao model (Dolev and Yao 1983) and supports many cryptographic primitives, including digital signature, symmetric and asymmetric encryption, hash function, and so on.

There're two types of channels in the formal model: a public channel for transmitting general protocol messages and private channel for transmitting smart card data between user and his smart card. The definition of these channels is given as below:

```
free cch: channel.
free sch: channel [private].
```

The variables and constants used in the protocol are defined as follows:

```
const P: bitstring.
const BIO_i: bitstring.
const pw_i: bitstring.
const x: bitstring.
free SK': bitstring [private].
free SK: bitstring [private].
```

The functions used in the protocol are defined as follows:

```
fun sco(bitstring, bitstring): bitstring.
fun Gen(bitstring): bitstring.
fun Rep(bitstring, bitstring): bitstring.
fun xor(bitstring, bitstring): bitstring.
fun mult(bitstring, bitstring): bitstring.
fun h(bitstring): bitstring.
```

Function sco, xor, mult, h represent bound symbol, exclusive or operation, scalar multiplication and hash function in the protocol, and function Gen and Rep are fuzzy extractor algorithms. The algebraic properties of these functions are modeled as the following equation and reduction:

```
equation forall m: bitstring, n: bitstring; xor(xor(m, n), n) = m.
In order to prove authentication, two events are defined as follows:
event UserAuthed(bitstring).
event UserStarted(bitstring).
```

The process part defines the action of participants and models the protocol as the parallel executions of them. According to the protocol, the following is the core message sequence for our protocol:

Message 1: User $U_i \rightarrow$ Server S: $REQUEST\{ID_i, D, F\}$
Message 2: Server $S \rightarrow$ User U_i: $CHALLENGE\{realm, Auth_s, H, t\}$
Message 3: User $U_i \rightarrow$ Server S: $RESPONSE\{ID_i, realm, Auth_u\}$

The actions of user U_i are composed of computing and then sending message 1 to S, waiting until he or she receives message 2 from S, computing and sending message 3 to S. We define user U_i as below:

```
let U_i =
        new a_i: bitstring;
        new ID_i: bitstring;
        let M = h(sco(a_i, pw_i)) in
        out(sch,(ID_i, M));
        in(sch, xR: bitstring);
        let (r: bitstring, s: bitstring) = Gen(BIO_i) in
        let B = xor(a_i, h(r)) in
        let C = h(sco(ID_i, sco(pw_i, a_i))) in
        !
        (
          event UserStarted(ID_i);
          new b: bitstring;
        new BIO_i': bitstring;
        let r' = Rep(BIO_i', s) in
        let a_i' = xor(B,h(r')) in
        let C' = h(sco(ID_i, sco(pw_i, a_i))) in
        if C' = C then
        let D = mult(b, P) in
        let D' = xor(xR, h(sco(a_i, pw_i))) in
        let F = h(sco(ID_i, sco(D, D'))) in
        out(cch, (ID_i, D, F));
        in (cch, (realm': bitstring, Auth_s': bitstring, H': bitstring, t': bitstring));
        let K' = mult(b, mult(D', H')) in
        let SK' = h(sco(ID_i, sco(t', K'))) in
        let xAuth_s = h(sco(D, sco(K', sco(D', sco(t', sco(SK', H')))))) in
        if xAuth_s = Auth_s' then
        let Auth_u = h(sco(ID_i, sco(realm', sco(K', sco(D', sco(t', sco(SK', sco(H', D)))))))) in
        out(cch, (ID_i, realm', Auth_u))
        ).
```

The actions of the server S are composed of receiving message 1 from U_i, computing and sending message 2 to U_i, waiting until he receives message 3 from U_i, and then verifying the message 3. We define the server as below:

```
let S =
        in(sch, (xID_i: bitstring, xM: bitstring));
        let R=xor(xM, h(sco(xID_i, x))) in
        out(sch, R);
        !
        (
         in(cch, (ID_i': bitstring, xD: bitstring, F': bitstring));
         let D'' = h(sco(ID_i', x)) in
         let xF = h(sco(ID_i',sco(xD,D''))) in
         if F' = xF then
         new u: bitstring;
         new t: bitstring;
         new realm: bitstring;
         let H = mult(u, P) in
         let K = mult(u,mult(D'',xD)) in
         let SK = h(sco(ID_i', sco(t, K))) in
         let Auth_s = h(sco(xD, sco(K, sco(D'', sco(t, sco(SK, H)))))) in
         out(cch, (realm, Auth_s, H, t));
         in(cch, (ID_i'': bitstring, realm'': bitstring, Auth_u': bitstring));
         if Auth_u'=h(sco(ID_i'',sco(realm'',sco(K,sco(D'',sco(t,sco(SK,sco(H,xD)))))))) then
         event UserAuthed(ID_i'');
         0
        ).
```

The protocol is defined as the parallel executions of the two participants:

```
process !U_i| S
```

In order to verify mutual authentication and the session key security, we define the following queries for checking the events' correspondence and the *attacker* queries respectively:

```
query id: bitstring; inj-event(UserAuthed(id)) ==> inj-event(UserStarted(id)).
query attacker(SK).
query attacker(SK').
```

The above code is performed in the latest version 1.90 of ProVerif to show that the correspondence query is true and the two attacker queries are not true. That is, the authentication property and security are satisfied, referring to the Fig. 1.

Security analysis

Session key security

Due to the impossibility of solving the computational Diffie-Hellman (CDH) problem, an adversary can neither know $h(ID_i \| x)$ nor compute ubP from bP and uP. That is, the adversary cannot compute the session key $SK = h(ID_i \| t \| uh(ID_i \| x)bP)$.

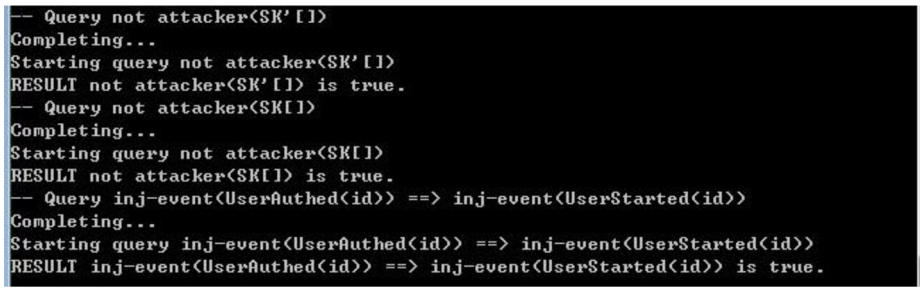

Fig. 1 Verification result of the protocol

Mutual authentication

The user U_i and the server S can authenticate each other by checking the correctness of F, $Auth_u$ and $Auth_s$, respectively. Without the knowledge of $h(ID_i\|x)$, no one except the user and the server can compute $Auth_u$ and $Auth_s$.

Replay attack

An adversary may intercept the request message $REQUEST\{ID_i, D, F\}$ and replay to the server, where $D = bP$, $D' = h(ID_i\|x)$ and $F = h(ID_i\|D\|D')$. Without the knowledge of b, he or she cannot generate the correct response message $RESPONSE\{ID_i, realm, Auth_u\}$ after receiving the server's message $CHALLENGE\{realm, Auth_s, H, t\}$. Then the server could detect the attack by checking the correctness of $Auth_u$. On the other hand, the adversary may intercept the challenge message $CHALLENGE\{realm, Auth_s, H, t\}$ and replay it to the user, where $K = uh(ID_i\|x)D$ and $Auth_s = h(D\|K\|D''\|t\|SK\|H)$. As the user generates a new $D = bP$ for each session, the attack can be detected by checking the correctness of $Auth_s$. Therefore, proposed SIP authentication scheme can resist the replay attack.

Off-line password guessing attack

Suppose that the adversary gets the data $\{B, C, \lambda, R\}$, where $B = a_i \oplus h(\eta)$, $C = h(ID_i\|pw_i\|a_i)$, $R = h(a_i\|pw_i) \oplus h(ID_i\|x)$. He could also eavesdrop the message $REQUEST\{ID_i, D, F\}$, $CHALLENGE\{realm, Auth_s, H, t\}$ and $RESPONSE\{ID_i, realm, Auth_u\}$ transmitted between U_i and S. The adversary may guess a password pw_i^*, but without the knowledge of S's secret key x, he or she can neither compute the random number a_i nor verify if his guessed password is correct or not. Hence, our scheme can resist the off-line password guessing attack.

For similar reasons, our protocol can resist smart card stolen attacks.

Privileged insider attack

In the registration phase of our scheme, U_i chooses the random number a_i, the password pw_i, and computes the hash value $h(a_i\|pw_i)$. Then U_i sends the hash value to S. The privileged insider can't get pw_i as it is protected by the random number a_i and the secure hash function.

Impersonation attack

Without the knowledge of S's secret key x, the attacker can neither generate the valid challenge message $CHALLENGE\{realm, Auth_s, H, t\}$, where $Auth_s = h(D\|K\|D''\|t\|SK\|H)$ and $K = uh(ID_i\|x)D$, nor compute the legal message $RESPONSE\{ID_i, realm, Auth_u\}$. Note that all messages are transmitted via a secure channel in registration phase, which are supposed to be free of corruption. So our scheme could withstand the impersonation attack.

Stolen-verifier attack

In the proposed scheme, S only needs to keep its key x secret. No password-verifier table is required to be stored in the server's database. Therefore, our scheme can resist the stolen-verifier attack.

Man-in-the-middle attack

From the above security analysis, we know that our scheme could provide mutual authentication between U_i and S, and can resist off-line password guessing attack and impersonation attack. Hence, our scheme is secure against the man-in-the-middle attack.

Perfect forward secrecy

In our protocol, the session key is $SK = h(ID_i\|t\|uh(ID_i\|x)bP)$, even if an adversary corrupts all secret parameters such as S's secret key x and U_i's password pw_i, he or she still cannot compute $uh(ID_i\|x)bP$ from bP and uP due to the intractability of CDH problem. Therefore, the introduced scheme can provide perfect forward secrecy.

Security and performance comparisons

Security and computation cost comparison

The security and computation cost comparisons between the proposed scheme and some related schemes (Zhang et al. 2014; Tu et al. 2015; Irshad et al. 2015; Arshad and Nikooghadam 2016; Farash 2016; Mishra et al. 2016; Chaudhry et al. 2015a; Wu et al. 2015) are given in Tables 2 and 3. For convenience, some notations are defined as follows: SY, H, MI, SM and PA are the operation times of a symmetric key encryption or decryption, hash function, modular inversion, scalar multiplication and point addition over elliptic curve, respectively.

Very recently, Kilinc and Yanik (2014) have estimated the complexity of various cryptographic operations by using the PBC library. The actual execution time for the above notations of operations are as follows: SY is about 0.0046 ms, H is about 0.0023 ms, MI is about 0.0056 ms (Koblitz et al. 2000), SM is about 2.226 ms, PA is about 0.0288 ms.

From Tables 2 and 3, we can conclude that our scheme enjoys better security than others, and higher efficiency than other related schemes except Mishra et al.'s protocol (Chaudhry et al. 2015a). Unfortunately, Mishra et al.'s protocol cannot provide perfect forward secrecy since the session key is

$$SK = h\big(username\big\|h(mk\big\|username\|N)\|(mk \cdot uP)_x\|T_2\|T_3\big),$$

Table 2 Security comparison

Schemes	Zhang et al. (2014)	Tu et al. (2015)	Irshad et al. (2015)	Arshad and Nikoogh-adam (2016)	Farash (2016)	Mishra et al. (2016)	Chaudhry et al. (2015a)	Wu et al. (2015)	Our scheme
Session key security	Y	Y	Y	Y	Y	Y	Y	Y	Y
Replay attack	Y	Y	Y	Y	Y	Y	Y	Y	Y
Perfect forward secrecy	Y	Y	Y	Y	Y	N	Y	Y	Y
Man-in-the-middle attack	Y	N	Y	Y	Y	Y	Y	Y	Y
Stolen-verifier attack	Y	Y	Y	Y	Y	Y	Y	Y	Y
Impersonation attack	N	N	N	N	N	Y	Y	Y	Y
Privileged insider attack	N	Y	Y	N	Y	Y	Y	N	Y
Mutual authentication	Y	Y	Y	Y	Y	Y	Y	Y	Y
Password guessing attack	Y	Y	Y	Y	N	Y	Y	Y	Y

Y the scheme can resist this attack or provide this property

N the scheme cannot resist this attack or cannot provide this property

Table 3 Computation cost comparison

Schemes	RP	LAAP	PCP	TC	AT (ms)
Zhang et al. (2014)	1SM + 2H	8SM + 2PA + 11H	1SM + 4SY + 6H	10SM + 2PA + 4SY + 19H	22.3797
Tu et al. (2015)	1SM + 2H	7SM + 1PA + 10H	1SM + 4SY + 6H	9SM + 1PA + 4SY + 18H	20.1226
Irshad et al. (2015)	1SM + 2H	7SM + 12H	1SM + 4SY + 6H	9SM + 4SY + 20H	20.0984
Arshad and Nikoogh-adam (2016)	2H	4SM + 8H + 1MI	9H	4SM + 19H + 1MI	8.9533
Farash (2016)	1SM + 2H	7SM + 1PA + 10H	1SM + 4SY + 6H	9SM + 1PA + 4SY + 18H	20.1226
Mishra et al. (2016)	4H	3SM + 12H	6H	3SM + 22H	7.184
Chaudhry et al. (2015a)	3H	6SM + 7H	3H	6SM + 13H	13.3859
Wu et al. (2015)	4H	4SM + 4SY + 12H	4H	4SM + 4SY + 20H	8.9684
Our scheme	4H	4SM + 12H	5H	4SM + 21H	8.9523

RP registration phase, *LAAP* login and authentication phase, *PCP* password change phase, *TC* total computation, *AT* actual time

where mk is the secret key of the server S, T_2 and T_3 are timestamps, u is nonce chosen by the user and N is registration sign. According to the definition of perfect forward secrecy, if an attacker can know the secret key mk of S then he or she can compute the

Table 4 Storage capacity comparison

Schemes	Zhang et al. (2014)	Tu et al. (2015)	Irshad et al. (2015)	Arshad and Nikooghadam (2016)	Farash (2016)	Mishra et al. (2016)	Chaudhry et al. (2015a)	Wu et al. (2015)	Our scheme
Memory needed in smart card (bits)	292	292	456	128	292	932	676	676	896

session key SK. Generally, we can use Diffie-Hellman key exchange algorithm to achieve perfect forward secrecy, but it needs more scalar multiplication operations over elliptic curve.

Storage capacity comparison

Since the proposed protocol is developed for applications using smart card, the memory requirement is a key parameter in concern. Therefore, we have also compared the storage capacity of our scheme with other related schemes (Zhang et al. 2014; Tu et al. 2015; Irshad et al. 2015; Arshad and Nikooghadam 2016; Farash 2016; Mishra et al. 2016; Chaudhry et al. 2015a; Wu et al. 2015). We assume that hash function outputs 256 bits, the size of a point on elliptic curve is 164 bits, the length of a random nonce is 128 bits, and the length of an identity is 128 bits. In the proposed scheme, the smart card needs to store $\{B, C, \lambda, R\}$ which is $256 + 256 + 128 + 256 = 896$ bits. The storage capacities of other relevant schemes have been shown in Table 4, which shows that the memory of smart cards needed in all schemes are less than 1 k bit.

Conclusions

In this paper, we propose a secure and efficient biometrics-based SIP authentication scheme. We apply formal verification tools and security analysis against various attacks to show that our proposed scheme achieves both security and authentication. Moreover, the performance evaluation validates that our scheme has very high efficiency in comparison to other related schemes.

Authors' contributions

Conceived and designed the experiments: QX. Performed the experiments: TZX. Analyzed the data: QX and TZX. Contributed reagents/materials/analysis tools: QX. Wrote the paper: QX and TZX. Designed the scheme and wrote the paper: QX and TZX. Verified the authentication and security of the proposed scheme in the latest version 1.9 of ProVerif: TZX. Both authors read and approved the final manuscript.

Author information

Qi Xie is a professor in Key Laboratory of Cryptography and Network Security, Hangzhou Normal University, China. He received his PhD degree in applied mathematics from Zhejiang University, China, in 2005. He was a visiting scholar between 2009 and 2010 at Department of Computer Science, University of Birmingham in UK, and a visiting scholar to the Department of Computer Science at City University of Hong Kong in 2012. His research area is applied cryptography, including digital signatures, authentication and key agreement protocols etc. He has published over 60 research papers in international journals and conferences, and served as co-chair of ISPEC 2012 and ASIACCS 2013. Zhixiong Tang is currently a M.S. candidate of Hangzhou Normal University, China. His research interests include authentication and key exchange protocols.

Acknowledgements
This research was supported by Natural Science Foundations of Zhejiang Province (No. LZ12F02005), and the Major State Basic Research Development (973) Program of China (No. 2013CB834205).

Competing interests
The authors declare that they have no competing interests.

Received: 26 January 2016 Accepted: 30 June 2016
Published online: 11 July 2016

References

Abadi M, Fournet C (2001) Mobile values, new names, and secure communication. In: Proceedings of the 28th ACM SIGPLAN-SIGACT symposium on principles of programming languages. ACM, New York, pp 104–115
Abadi M, Blanchet B, Comon-Lundh H (2009) Models and proofs of protocol security: a progress report. Computer aided verification, vol 5643. Springer, Heidelberg, pp 35–49
Armando A, Basin D, Boichut Y, Chevalier Y, Compagna L, Cuellar J et al (2005) The AVISPA tool for the automated validation of internet security protocols and applications. Computer aided verification, vol 3576. Springer, Heidelberg, pp 281–285
Arshad R, Ikram N (2013) Elliptic curve cryptography based mutual authentication scheme for session initiation protocol. Multimed Tools Appl 66(2):165–178
Arshad H, Nikooghadam M (2016) An efficient and secure authentication and key agreement scheme for session initiation protocol using ECC. Multimed Tools Appl 75(1):181–197
Burrows M, Abadi M, Needham RM (1989) A logic of authentication. ACM Trans Comput Syst 8(1):18–36
Chaudhry SA, Mahmood K, Naqvi H, Khan MK (2015a) An improved and secure biometric authentication scheme for telecare medicine information systems based on elliptic curve cryptography. J Med Syst 39(11):1–12
Chaudhry SA, Naqvi H, Sher M, Farash MS, Hassan MU (2015b) An improved and provably secure privacy preserving authentication protocol for SIP. Peer-to-Peer Netw Appl. doi:10.1007/s12083-015-0400-9
Dodis Y, Reyzin L, Smith A (2004) Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. Advances in cryptology-Eurocrypt 2004, vol 3027. Springer, Heidelberg, pp 523–540
Dolev D, Yao AC (1983) On the security of public key protocols. IEEE Trans Inf Theory 29(2):198–208
Durlanik A, Sogukpinar I (2005) SIP authentication scheme using ECDH. World Enformatika Soc Trans Eng Comput Technol 8:350–353
Farash MS (2016) Security analysis and enhancements of an improved authentication for session initiation protocol with provable security. Peer-to-Peer Netw Appl 9(1):82–91
Farash MS, Attari MA (2013) An enhanced authenticated key agreement for session initiation protocol. Inf Technol Control 42(4):333–342
Franks J, Hallam-Baker PM, Hostetler JL, Lawrence SD, Leach PJ, Luotonen A, Stewart LC (1999) HTTP authentication: basic and digest access authentication. IETF RFC 2617
He D, Chen J, Chen Y (2012) A secure mutual authentication scheme for session initiation protocol using elliptic curve cryptography. Secur Commun Netw 5(12):1423–1429
Huang HF, Wei WC (2006) A new efficient authentication scheme for session initiation protocol. Computing 1(2):1–3
Irshad A, Sher M, Faisal MS, Ghani A, Hassan MU, Ashraf ChS (2014) A secure authentication scheme for session initiation protocol by using ECC on the basis of the Tang and Liu scheme. Secur Commun Netw 7(8):1210–1218
Irshad A, Sher M, Rehman E, Ch SA, Hassan MU, Ghani A (2015) A single round-trip SIP authentication scheme for Voice over Internet Protocol using smart card. Multimed Tools Appl 74(11):3967–3984
Jo H, Lee Y, Kim M, Kim S, Won D (2009) Off-line password-guessing attack to Yang's and Huang's authentication schemes for session initiation protocol. In: Fifth international joint conference on INC, IMS and IDC, IEEE, Seoul, 25–27 Aug 2009
Kilinc HH, Yanik T (2014) A survey of SIP authentication and key agreement schemes. IEEE Commun Surv Tutor 16(2):1005–1023
Koblitz N, Menezes A, Vanstone S (2000) The state of elliptic curve cryptography. Des Code Crypt 19(2):173–193
Kumari S, Chaudhry SA, Wu F, Li X, Farash MS, Khan MK (2015) An improved smart card based authentication scheme for session initiation protocol. Peer-to-Peer Netw Appl. doi:10.1007/s12083-015-0409-0
Li CT, Hwang MS (2010) An efficient biometrics-based remote user authentication scheme using smart cards. J Netw Comput Appl 33(1):1–5
Liu F, Koenig H (2011) Cryptanalysis of a SIP authentication scheme. Communications and multimedia security, vol 7025. Springer, Heidelberg, pp 134–143
Mishra D, Das AK, Mukhopadhyay S (2016) A secure and efficient ECC-based user anonymity-preserving session initiation authentication protocol using smart card. Peer-to-Peer Netw Appl 9(1):171–192
Pu Q, Wang J, Wu S (2013) Secure SIP authentication scheme supporting lawful interception. Secur Commun Netw 6(3):340–350
Tang H, Liu X (2013) Cryptanalysis of Arshad et al.'s ECC-based mutual authentication scheme for session initiation protocol. Multimed Tools Appl 65(3):321–333
Tsai JL (2009) Efficient nonce-based authentication scheme for session initiation protocol. Int J Netw Secur 9(1):12–16
Tu H, Kumar N, Chilamkurti N, Rho S (2015) An improved authentication protocol for session initiation protocol using smart card. Peer-to-Peer Netw Appl 8(5):903–910
Witteman M (2002) Advances in smartcard security. Inf Secur Bull 7(2002):11–22

Wu L, Zhang Y, Wang F (2009) A new provably secure authentication and key agreement protocol for SIP using ECC. Comput Stand Inter 31(2):286–291
Wu K, Gong P, Wang J, Yan X, Li P (2013) An improved authentication protocol for session initiation protocol using smart card and elliptic curve cryptography. Rom J Inf Sci Technol 16(4):324–335
Wu F, Xu L, Kumari S, Li X (2015) A novel and provably secure biometrics-based three-factor remote authentication scheme for mobile client–server networks. Comput Electr Eng 45:274–285
Xie Q (2012) A new authenticated key agreement for session initiation protocol. Int J Commun Syst 25(1):47–54
Yang D, Yang B (2009) A new password authentication scheme using fuzzy extractor with smart card. 2009 International conference on computational intelligence and security, vol 2. IEEE, Beijing, pp 278–282
Yang CC, Wang RC, Liu WT (2005) Secure authentication scheme for session initiation protocol. Comput Secur 24(5):381–386
Yoon EJ, Shin YN, Jeon IS, Yoo KY (2010a) Robust mutual authentication with a key agreement scheme for the session initiation protocol. IETE Tech Rev 27(3):203–213
Yoon EJ, Yoo KY, Kim C, Hong YS, Jo M, Chen HH (2010b) A secure and efficient SIP authentication scheme for converged VoIP networks. Comput Commun 33(14):1674–1681
Zhang L, Tang S, Cai Z (2014) Efficient and flexible password authenticated key agreement for voice over internet protocol session initiation protocol using smart card. Int J Commun Syst 27(11):2691–2702