Peer-to-Peer Netw. Appl. (2016) 9:82–91
DOI 10.1007/s12083-014-0315-x

Security analysis and enhancements of an improved authentication for session initiation protocol with provable security

Mohammad Sabzinejad Farash

Received: 11 April 2014 / Accepted: 6 October 2014 / Published online: 16 October 2014

Abstract Very recently, Tu et al. proposed an authentication scheme for session initiation protocol using smart card to overcome the security flaws of Zhang et al.'s protocol. They claimed that their protocol is secure against known security attacks. However, in this paper, we indicate that Tu et al.'s protocol is insecure against impersonation attack. We show that an adversary can easily masquerade as a legal server to fool users. As a remedy, we also improve Tu et al.'s protocol without imposing extra computation cost. To show the security of our protocol, we prove its security in the random oracle model.

Keywords Password-based protocol · Elliptic curve · Session initiation protocol · Smart card · Random oracle model

1 Introduction

The session initiation protocol (SIP) is an application layer signaling protocol for creating, modifying, and terminating multimedia sessions among one or more participants [1]. SIP was developed by the Internet Engineering Task Force (IETF) in 1996. With the widespread application of the Voice over IP (VoIP) in Internet [2–4] and mobility management [5–8], SIP has been receiving a lot of attention and the security of SIP is becoming increasingly important [9]. 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 ((e.g., [10–24])), especially based on Elliptic Curve Cryptography (ECC), have been proposed to provide security for SIP for a decade [25–32].

In 2005, Yang et al. [33] 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.[34] 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 [35], 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 et al. [36] 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 et al.'s scheme reduced the total execution time and the requirements for memory in comparison with Yang et al.'s scheme. However, Yoon et al. [37] indicated that Durlanik et al.'s scheme still suffered from off-line password guessing and Denning-Sacco attacks, and projected an improved scheme to overcome the weaknesses. However, Liu et al. [38] demonstrated that Yoon et al.'s scheme still puts up with off-line password guessing and insider attacks.

In 2009, Tsai [39] 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 equipments. However, it was still defenseless 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

M. S. Farash (✉)
Faculty of Mathematical Sciences and Computer, Kharazmi University, Tehran, Iran
e-mail: sabzinejad@khu.ac.ir

(PFS) [40–42]. To deal with the problems, Arshad et al. proposed an ECC-based authentication scheme [42]. But, Tang et al. [43] demonstrated the vulnerability of Arshad et al.'s scheme to off-line password guessing attack and introduced an improved scheme to overcome the weakness.

In 2010, Yoo et al. [44] also proposed an authentication scheme based on ECC to deal with the problems in Tsai et al.'s scheme. In 2012, Xie [45] pointed out that Yoo et al.'s scheme still suffers from stolen-verifier and off-line password guessing attacks and proposed an improved scheme. However, Farash and Attari [46] show that Xie's scheme is also insecure and proposed an enhanced scheme. Recently, Zhang et al. [47] proposed a new password-based authenticated protocol, but Tu et al. [48] foun out thet it is insecure against impersonation attacks. Tu et al. proposed an improved authentication protocol for session initiation protocol using smart card to overcome the security flaws of Zhang et al.'s protocol. They claimed that their protocol satisfies all the security requirements for such protocols. However, this paper indicates that Tu et al.'s protocol is also vulnerable to impersonation attack. To remedy this problem, we proposed an improved protocol by taking a slight change in Tu et al.'s protocol. The security of the improved protocol is proved in the random oracle model.

The rest of this paper is organized as follows. We review Tu et al.'s protocol in Section 2. In Section 3, we propose the security weaknesses of Zhang et al.'s protocol. Our improved protocol and its security proof are proposed in Sections 4 and 5, respectively. A comparison between our improved protocol and the related protocols is proposed in Section 6. Finally, we conclude our paper in Section 7.

2 Review of Tu et al.'s protocol

In this Section, we review Tu et al.'s password-based authenticated key agreement protocol [48] using the notations shown in Table 1). This protocol has four phases: setup, registration, authentication, and password changing phases.

2.1 Setup phase

In this phases, the server chooses the following items:

- The elliptic curve E over the finite field F_q,
- the additive group $\mathbb{G}$ generated by the base point P with the prime order p,
- three one-way hash functions $h : \{0,1\}^* \rightarrow \{0,1\}^k$, $h_1 : \mathbb{G} \times \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^k$, and $h_2 : \mathbb{G} \times \mathbb{G} \times \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^k$, and
- the random number $s \in \mathbb{Z}_p^*$ as the server's secret key.

Finally, the server publishes the public parameters $\{E(F_q), P, p, \mathbb{G}, h, h_1, h_2\}$, and maintains the secret key s.

2.2 Registration phase

In this phase, the user U who wants to become a legal user of a remote server performs the following steps over a secure channel:

- U freely chooses the password PW_U and the random number $a_U \in \mathbb{Z}_p^*$, computes $h(PW_U \| a_U)$, and sends the messages $\{h(PW_U \| a_U), username_U\}$ to the remote server.
- After receiving the message $\{h(PW_U \| a_U), username_U\}$, the server computes $R_U = (h(PW_U \| a_U) + h(username_U || s))P$, stores R_U in a smart card, and finally delivers the smart card to U.
- Upon receiving the smart card, U inserts the random numbers a_U in the memory of the smart card and memorizes the password PW_U in his/her mind.

2.3 Authentication phase

When the user U wants to login to the remote server, he/she inserts his/her smart card to a card reader and inputs his/her username and password PW_U. Then, the smart card and the remote server perform as follows, shown in Fig. 1:

Step A1. The smart card randomly chooses $b \in \mathbb{Z}_p^*$ and computes

$$V = bP,$$
$$V' = b(R - h(PW_U \| a_U)P),$$
$$W = h(username_U || V || V').$$

The smart card then sends $REQUEST\{username_U, V, W\}$ to the remote server.

Step A2. Upon receiving $REQUEST\{username_U, V, W\}$, the remote server firstly computes $V'' = h(username_U || s)V$ and $W' = h(username_U || V || V'')$, then it checks if $W = W'$. If it holds, the remote server selects the random numbers $c, r \in \mathbb{Z}_p^*$ and computes $C = cP$, $K = cV = cbP$, $SK = h_1(K \| r \| username_U)$ and $Auth_s = h_2(K \| W' \| r \| SK)$. Finally, the remote server sends the message $CHALLENGE\{realm, Auth_s, C, r\}$ to U.

Step A3. Upon receiving the message $CHALLENGE\{realm, Auth_s, C, r\}$, U computes $K = bC = bcP$ and $SK = h_1(K \| r \| username_U)$. Then he/she verifies $Auth_s = h_2(K \| W \| r \| SK)$. If it holds, the smart card computes $Auth_u = h_2(K \| W \| r + 1 \| SK)$ and sends the message $RESPONSE\{realm, Auth_u\}$ to the remote server.

Step A4. Upon receiving the message $RESPONSE\{realm, Auth_u\}$, the remote server checks if $Auth_u = h_2(K \| W' \| r + 1 \| SK)$. If it holds, the remote server confirms that the claimant U is a legal user.

Table 1 The notations

Notation	Description
U	a user
$username_A$	the unique identity of the user A
PW_U	the password of the user U
(R_U, a_U)	the secret information of the user U stored in the smart card
p, q	two prime numbers
E	an elliptic curve
F_q	a finite field
$E(F_q)$	a group contains the points on the elliptic curve E over the finite field F_p
P	an element of $E(F_q)$ with the prime order p
$\mathbb{G}$	a subgroup of $E(F_p)$ generated by the base point P
$\mathbb{Z}_p^*$	the non-zero integers modulus p
h	the hash function $h : \{0,1\}^* \rightarrow \{0,1\}^k$
h_1	the hash function $h_1 : \mathbb{G} \times \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^k$
h_2	the hash function $h_2 : \mathbb{G} \times \mathbb{G} \times \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^k$
Enc, Dec	symmetric encryption and decryption algorithms
s	the secret key of the server

2.4 Password changing phase

The user U can change his/her password freely in this phase. To do so, he/she firstly executes the login and authentication phase with his/her $username_U$ and the old password PW_U. After receiving the successful authentication and sharing the session key SK, the user U does as follows:

Step C1. U freely selects the new password PW_U^* and the random number $N, a_U^* \in \mathbb{Z}_p^*$. U then computes $C_1 =$

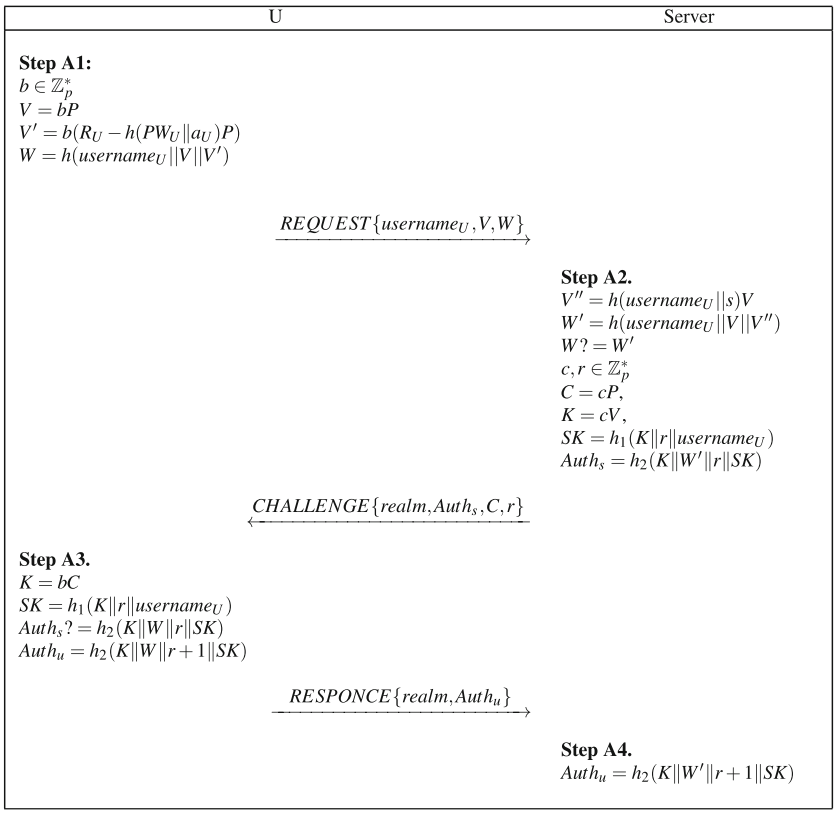

Fig. 1 The authentication phase of Tu et al.'s protocol [48]

$Enc_{SK}(username_U \| N \| h(PW_U^* \| a_U^*) \| h(username_U \| N \| h(PW_U^* \| a_U^*)))$. Next, U sends $\{username_U, C_1, N\}$ to the server.

Step C2. Upon receiving the message $\{username_U, C_1, N\}$, the server decrypts C_1 and verifies the integrity of $h(username_U \| N \| h(PW_U^* \| a_U^*))$. If it is valid, the server computes $R_U^* = h(PW_U^* \| a_U^*) - h(username_U || s)P$, encrypt it as $C_2 = Enc_{SK}(R_U^* \| h(username_U \| N + 1 \| R_U^*))$, and sends C_2 to U.

Step C3. Upon receiving the message, U decrypts the message and checks the integrity of it. If it is valid, U stores $(PW_U^* \| a_U^*)$ in the smart card.

3 Cryptanalysis and improvement of Tu et al.'s protocol

In this section, we find out that an active adversary can mount an impersonation attack on Tu et al.'s protocol [48]. We show that an active attacker can masquerade as the remote server to make a session key with users. The details of this attack, shown in Fig. 2, are as follows:

Step I1. When the user U wants to login to the remote server, by computing

$$V = bP, \tag{1}$$
$$V' = b(R - h(PW_U \| a_U)P), \tag{2}$$
$$W = h(username_U || V || V'), \tag{3}$$

and sending the request message $REQUEST\{username_U, V, W\}$ to the server, the attacker $\mathscr{A}$ intercepts and records it.

Step I2. $\mathscr{A}$ then selects random numbers $c, r \in \mathbb{Z}_p^*$, computes

$$C = cP, \tag{4}$$
$$K = cV = cbP, \tag{5}$$
$$SK = h_1(K \| r \| username_U), \tag{6}$$
$$Auth_s = h_2(K \| W \| r \| SK), \tag{7}$$

and sends the message $CHALLENGE\{realm, Auth_s, C, r\}$ to U.

Step I3. Upon receiving the message $CHALLENGE\{realm, Auth_s, C, r\}$, U computes

$$K' = bC = bcP, \tag{8}$$
$$SK' = h_1(K' \| r \| username_U). \tag{9}$$

Then he/she verifies

$$Auth_s = h_2(K' \| W \| r \| SK'). \tag{10}$$

If it holds, U believes that the received message was generated by the legal server. Then U computes $Auth_u = h_2(K' \| W \| r + 1 \| SK')$ and sends the message $RESPONSE\{realm, Auth_u\}$ to the remote server.

Step I4. $\mathscr{A}$ intercepts the response message $RESPONSE\{realm, Auth_u\}$.

Proposition 1 *At the end of the proposed impersonation attack, the attacker $\mathscr{A}$ has accepted as the legal server by the user U.*

Proof As mentioned in Step I3, U ensures that the received message was generated by the legal server if the Eq. 10 holds. In the other hand, the Eq. 10 holds if the $K = K'$ and $SK' = SK$. According to the Eqs. 1, 4, 5 and 8, it is clear that K and K' are same as follows:

$$\begin{aligned} K' &= bC \\ &= bcP \\ &= cV \\ &= K. \end{aligned}$$

Resultantly, in respect to the Eqs. 6 and 9, the equality $K = K'$, SK' and SK are equal as follows:

$$\begin{aligned} SK' &= h_1(K' \| r \| username_U) \\ &= h_1(K \| r \| username_U) \\ &= SK. \end{aligned}$$

Thus, the verification Eq. 10 holds as follows:

$$\begin{aligned} h_2(K' \| W \| r \| SK') &= h_2(K \| W \| r \| SK) \\ &= Auth_s. \end{aligned}$$

Therefore, $\mathscr{A}$ has succeeded to masquerade as the remote server and share the session key $SK = h_1(K \| r \| username_U)$ with U. □

4 Our improvement

The security flaw of Tu et al.'s protocol [48] is due to this fact that the value of $Auth_s$ is computed using the public parameters W, V and $username_U$, and the random numbers r and c. A straightforward solution to overcome the problem is to use a secret parameter for computing $Auth_s$. As can be seen in the original protocol, the parameter $V' = b(R - h(PW_U \| a_U)P) = h(username_U || s)V = V''$ is a secret parameter for both the user and the remote server. Thus, we improve the the parameter $Auth_s$ as $h_2(K \| V'' \| r \| SK)$ for the server side. Therefore, if the adversary $\mathscr{A}$ wants to impersonate the remote server, he can not compute $Auth_s$

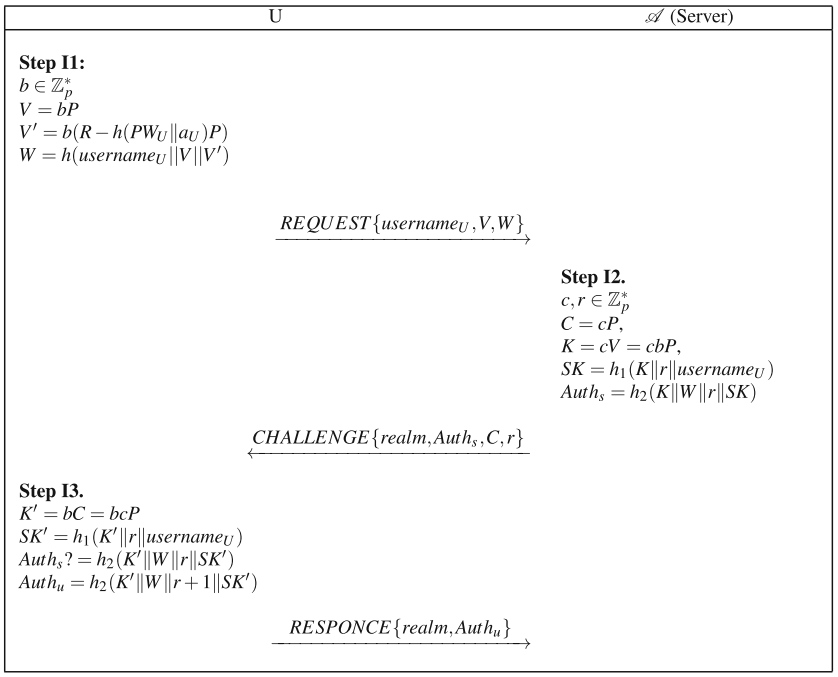

Fig. 2 The server impersonation attack on Tu et al.'s protocol

since V'' is unknown for him. Note that, this improvement does not impose extra cost to the original protocol.

The details of our improvement on the authentication phase of Tu et al.'s protocol, outlined in Fig. 3, are as follows:

When the user U wants to login to the remote server, he/she inserts his/her smart card to a card reader and inputs his/her username and password PW_U. Then, the smart card and the remote server perform as follows, shown in Fig. 1:

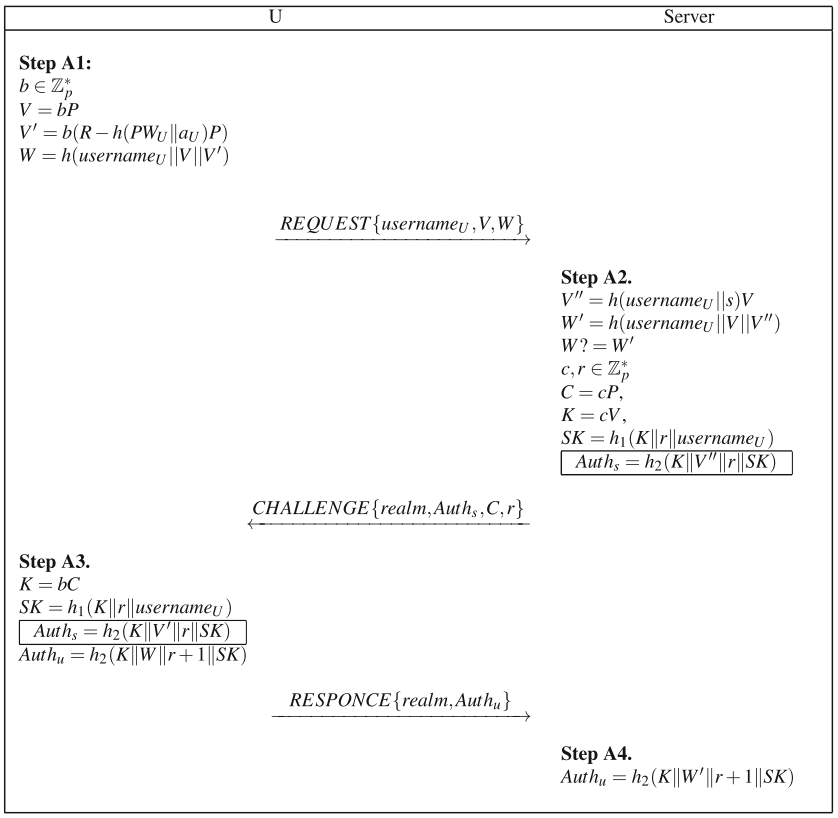

Fig. 3 Our improved protocol

Step A1. The smart card randomly chooses $b \in \mathbb{Z}_p^*$ and computes

$$V = bP, \tag{11}$$
$$V' = b(R - h(PW_U \| a_U)P), \tag{12}$$
$$W = h(username_U || V || V'). \tag{13}$$

The smart card then sends $REQUEST\{username_U, V, W\}$ to the remote server.

Step A2. Upon receiving $REQUEST\{username_U, \{V, W\}$, the remote server firstly computes

$$V'' = h(username_U || s)V, \tag{14}$$
$$W' = h(username_U || V || V''). \tag{15}$$

then it checks if $W = W'$. If it holds, the remote server selects the random numbers $c, r \in \mathbb{Z}_p^*$ and computes

$$C = cP, \tag{16}$$
$$K = cV, \tag{17}$$
$$SK = h_1(K \| r \| username_U), \tag{18}$$
$$Auth_s = h_2(K \| V'' \| r \| SK). \tag{19}$$

Finally, the remote server sends the message $CHALLENGE\{realm, Auth_s, C, r\}$ to U.

Step A3. Upon receiving the message $CHALLENGE\{realm, Auth_s, C, r\}$, U computes

$$K = bC, \tag{20}$$
$$SK = h_1(K \| r \| username_U). \tag{21}$$

Then he/she verifies

$$Auth_s = h_2(K \| W \| r \| SK). \tag{22}$$

If it holds, the smart card computes

$$Auth_u = h_2(K \| W \| r + 1 \| SK), \tag{23}$$

and sends the message $RESPONSE\{realm, Auth_u\}$ to the remote server.

Step A4. Upon receiving the message $RESPONSE\{realm, Auth_u\}$, the remote server checks if

$$Auth_u = h_2(K \| V' \| r + 1 \| SK). \tag{24}$$

If it holds, the remote server confirms that the claimant U is a legal user.

5 Security analysis of the improved protocol

In this section, we show that our protocol is secure in the random oracle model. We start with the formal security model and the algorithm assumption that will be used in our proof.

5.1 Security model

In order to make our scheme resist the known attacks to the authentication protocols, we use the method of provable security. The security proof is based on the model proposed by Abdalla and Pointcheval [49]. The model that we use is as follows:

5.1.1 Participants

An authentication protocol Π runs in a network of a number of interconnected participants where each participant is either a client $U \in \mathscr{U}$ or a trusted server $S \in \mathscr{S}$. The set $\mathscr{S}$ is assumed to involve only a single server for simplicity. Each of the participants may have several instances called oracles involved in distinct executions of the protocol Π. We refer to i-th instance of U (resp. S) in a session as Π_U^i (resp. Π_S^i). Every instance Π_U^i (resp. Π_S^j) has a partner ID pid_U^i (resp:pid_S^j), a session ID sid_U^i (resp:sid_S^j), and a session key sk_U^i (resp:sk_S^j). pid_U^i (resp:pid_S^j) denotes the set of the identities that are involved in this instance. sid_U^i (resp:sid_S^j) denotes the flows that are sent and received by the instance Π_U^i (resp. Π_S^j). An instance Π_U^i (resp. Π_S^i) is said to be *accepted* if it holds a session key sk_U^i (resp:sk_S^j), a session identifier sid_U^i (resp:sid_S^j), and a partner identifier pid_U^i (resp:pid_S^j). Two instances Π_U^i and Π_S^j are considered *partnered* if and only if (1) both of them have accepted, (2) $pid_U^i = pid_S^j$, (3) $sid_U^i = sid_S^j$, (4) $sk_U^i = sk_S^j$.

5.1.2 Long-lived keys

Each client $U \in \mathscr{U}$ holds a password pw_U. Each server $S \in \mathscr{S}$ holds a vector $pw_S = \langle pw_U \rangle_{U \in \mathscr{U}}$ with an entry for each client.

5.1.3 Adversary model

The communication network is assumed to be fully controlled by an adversary $\mathscr{A}$, which schedules and mediates the sessions among all the parties. The adversary $\mathscr{A}$ is allowed to issue the following queries in any order:

Execute(Π_U^i, Π_S^j): This query models passive attacks in which the attacker eavesdrops on honest executions among the client instance Π_U^i and trusted server instance Π_S^j. The output of this query consists of the messages that were exchanged during the honest execution of the protocol Π.

SendClient(Π_U^i, m): The adversary makes this query to intercept a message and then modify it, create a new one,

or simply forward it to the client instance Π_U^i. The output of this query is the message that the client instance Π_U^i would generate upon receipt of message m. Additionally, the adversary is allowed to initiate the protocol by invoking SendClient(Π_U^i, $Start$).

SendServer(Π_S^i, m): This query models an active attack against a server. The adversary makes this query to obtain the message that the server instance Π_S^i would generate on receipt of the message m.

Reveal(Π_U^i): This query models the known session key attack. The adversary makes this query to obtain the session key of the instance Π_U^i.

Corrupt(U): This query returns to the adversary the long-lived key pw_U for participant U.

Test(Π_U^i): Only one query of this form is allowed to be made by the adversary to a fresh oracle. To respond to this query, a random bit $b \in \{0, 1\}$ is selected. If $b = 1$, then the session key held by Π_U^i is returned. Otherwise, a uniformly chosen random value is returned.

5.1.4 Fresh oracle

An oracle Π_U^i is called fresh if and only if the following conditions hold: (1) Π_U^i has accepted, and (2) Π_U^i or its partner (if exists) has not been asked a Reveal query after their acceptance.

5.1.5 Protocol Security

The security of an authentication protocol Π is modeled by the game $Game(\Pi, \mathscr{A})$. When playing this game, the adversary $\mathscr{A}$ can make many queries mentioned earlier to Π_U^i and Π_S^j. If $\mathscr{A}$ asks a single query, Test(Π_U^i), where Π_U^i has *accepted* and is *fresh*, then $\mathscr{A}$ outputs a single bit b'. The aim of $\mathscr{A}$ is correctly guessing the bit b in the test session. More precisely, we define the advantage of $\mathscr{A}$ as follows:

$$Adv_{\Pi,D}(\mathscr{A}) = |2Pr[b' = b] - 1|.$$

The protocol Π is said to be secure if $Adv_{\Pi,D}(\mathscr{A})$ is negligible.

5.2 Computational assumption

We define the decisional Diffie-Hellman (DDH) assumption which we use in the security proof of our scheme.

Definition 1 The DDH assumption can be precisely defined by two experiments, $Exp_{P,p}^{ddh-real}(W)$ and $Exp_{P,p}^{ddh-rand}(W)$. An adversary W is provided with uP, vP and uvP in the experiment $Exp_{P,p}^{ddh-real}(W)$, and uP, vP and wP in the experiment $Exp_{P,p}^{ddh-rand}(W)$, where u, v and w are drawn at random from $\mathbb{Z}_p^*$. Define the advantage of W in violating the DDH assumption, $Adv_{P,p}^{ddh}(W)$, as follows:

$$Adv_{P,p}^{ddh}(W) = \max\{|Pr[Exp_{P,p}^{ddh-real}(W) = 1] - Pr[Exp_{P,p}^{ddh-rand}(W) = 1]|\}.$$

5.3 Security proof

Theorem 1 *Let D be a uniformly distributed dictionary of passible passwords with size $|D|$. Let Π describes the improved authentication protocol defined in Fig. 3. Suppose that DDH assumption holds, Then,*

$$Adv_{\Pi,D}(\mathscr{A}) \le \frac{q_h^2 + q_{h1}^2 + q_{h2}^2}{2^k} + \frac{(q_s + q_e)^2}{p^2} + 2q_e \cdot Adv_{P,p}^{DDH}(W) + 2\max\{\frac{q_{h1}}{2^k}, \frac{q_s}{|D|}\},$$

where q_s denotes the number of Send queries; q_e denotes the number of Execute queries; q_h, q_{h1} and q_{h2} denotes the number of hash queries to h, $h1$ and $h2$, respectively.

Proof This proof consists of a sequence of hybrid games, starting at the real attack G_0 and ending up at game G_4 where the adversary has no advantage. For each game $G_i (0 \le i \le 4)$, we define $Succ_i$ as the event that $\mathscr{A}$ correctly guesses the bit b in the test session.

Game G_0. This game is the real protocol, in the random-oracle model. In this game, all the instances of U and the trusted server S are modeled as the real execution in the random oracle. By definition of event $Succ_i$, which means that the adversary correctly guesses the bit b involved in the Test-query, we have

$$Adv_{\Pi,D}(\mathscr{A}) = 2|\Pr[Succ_0] - \frac{1}{2}|. \quad (25)$$

Game G_1. This game is as the same as the game G_0 except that we simulate the hash oracles h, h_1 and h_2 as usual by maintaining hash lists h_{List}, $h1_{List}$ and $h2_{List}$ with entries of the form (Inp, $Outp$). On hash query for which there exists a record (Inp, $Outp$) in the hash list, return $Outp$. Otherwise, randomly choose $Outp \in \{0, 1\}^k$, send it to $\mathscr{A}$ and store the new tuple (Inp, $Outp$) into the hash list. We also simulate all the instances, as the real players would do, for the Send-query and for the Execute, SendClient, SendServer, Reveal, Corrupt and Test queries. From the viewpoint of the adversary, we easily see that the game is perfectly indistinguishable from the real attack. Hence,

$$\Pr[Succ_1] = \Pr[Succ_0]. \quad (26)$$

Game G_2. In this game, we simulate all the oracles in game G_1, except we cancel the game in which some

collisions appear on the partial transcripts (V, C) and on hash values. According to the birthday paradox, the probability of collisions in output of hash oracles are at most $q_h^2/2^{k+1}$, $q_{h1}^2/2^{k+1}$ and $q_{h1}^2/2^{k+1}$ where q_h, q_{h1} and q_{h2} denote the maximum number of hash queries. Similarly, the probability of collisions in the transcripts is at most $(q_s + q_e)^2/(2p^2)$, where q_s represents the number of queries to the SendClient and SendServer oracles and q_e represents the number of queries to the Execute oracle. So we have

$$|\Pr[Succ_2] - \Pr[Succ_1]| \leq \frac{q_h^2 + q_{h1}^2 + q_{h2}^2}{2^{k+1}} + \frac{(q_s + q_e)^2}{2p^2}. \quad (27)$$

Game G_3. In this game, we change the simulation of queries to the SendClient oracle. First, we randomly select a session executed by partner instances Π_U^i and Π_S^j.

- When SendClient$(\Pi_U^i, Start)$ is asked, we choose random values $u \in [1, p+1]$ and compute $V = uP$, $V' = u(R_U - h(PW_U \| a_U)P$ and $W = h(username_U || V || V')$, and return $\{username_U, V, W\}$ to $\mathscr{A}$.
- When SendClient$(\Pi_U^i, (username_U, V, W)$ is asked, we choose random values $v, r \in [1, p+1]$ and compute $V = vP$, SK and $Auth_s$ like the real protocol and return $\{realm, Auth_s, C, r\}$ to $\mathscr{A}$.

So, it can be easily seen that this game is perfectly indistinguishable from the previous game G_2. Hence,

$$\Pr[Succ_3] = \Pr[Succ_2]. \quad (28)$$

Game G_4. In this game, we once again change the simulation of queries to the SendClient oracle for the selected session in game G_3. This time, we change the way we compute K so that it become independent of password and ephemeral keys. When SendServer $(\Pi_S^j, (username_U, V, W))$ and SendClient $(\Pi_A^i, (realm, Auth_s, C, r))$ are asked, we set $K = wP$, where w is selected from $\mathbb{Z}_p^*$ at random. The difference between the game G_4 and the game G_3 is as follows:

$$|\Pr[Succ_4] - \Pr[Succ_3]| \leq q_e \cdot Adv_{P,p}^{DDH}(W). \quad (29)$$

By assuming a successful adversary $\mathscr{A}$ to distinguish G_3 and G_4, we construct a DDH solver W.

In game G_4, the Diffie-Hellman key K is random and independent with the user's password and ephemeral keys. So, there are three possible cases where the adversary distinguishes the real session key SK and the random key as follows:

Case 1. the adversary queries $(K, r, username_U)$ to h_1. The probability that this event occurs is $q_{h1}/2^k$.

Case 2. the adversary asks SendClient query except SendClient(Π_S^j, m) and successfully impersonates U to S. The adversary is not allowed to reveal static key PW_U of U. Thus, in order to impersonate A, the adversary has to obtain some information of the password PW_A of A. The probability is $1/D$. Since there are at most q_s sessions of this kind, the probability that this event occurs is lower than $q_s/|D|$

As a conclusion,

$$\Pr[Succ_4] = \frac{1}{2} + \max\{\frac{q_{h1}}{2^k}, \frac{q_s}{|D|}\}. \quad (30)$$

Combining the Eqs. 25, 26, 27, 28, 29 and 30 one gets the announced result as follows:

$$\begin{aligned}
Adv_{\Pi,D}(\mathscr{A}) &= 2|\Pr[Succ_0] - \frac{1}{2}| \\
&= 2|\Pr[Succ_0] - \Pr[Succ_4] + \max\left\{\frac{q_{h1}}{2^k}, \frac{q_s}{|D|}\right\}| \\
&\leq 2(|\Pr[Succ_0] - \Pr[Succ_4]| + \max\left\{\frac{q_{h1}}{2^k}, \frac{q_s}{|D|}\right\}) \\
&\leq 2(|\Pr[Succ_1] - \Pr[Succ_2]| + |\Pr[Succ_3] \\
&\quad - \Pr[Succ_4]| + \max\left\{\frac{q_{h1}}{2^k}, \frac{q_s}{|D|}\right\}) \\
&\leq \frac{q_h^2 + q_{h1}^2 + q_{h2}^2}{2^k} + \frac{(q_s + q_e)^2}{p^2} + 2q_e \cdot Adv_{P,p}^{DDH}(W) \\
&\quad + 2\max\{\frac{q_{h1}}{2^k}, \frac{q_s}{|D|}\}.
\end{aligned}$$

□

6 Computation comparison

To estimate the computation cost of our scheme, the following notations are defined: PM is the time complexity of scalar point multiplication, PA is the time complexity of elliptic curve point addition, H is time complexity of hash function and I is time complexity of modular inversion. The

Table 2 Computation comparisons

	User's computation	Server's computation	Total computation
Zhang et al.'s protocol [47]	$4PM + 1PA + 6H$	$4PM + 1PA + 5H + 1I$	$8PM + 2PA + 11H + 1I$
Tu et al.'s protocol [48]	$4PM + 1PA + 5H$	$3PM + 5H$	$7PM + 1PA + 10H$
Improved protocol	$4PM + 1PA + 5H$	$3PM + 5H$	$7PM + 1PA + 10H$

computation cost of the proposed protocol and a comparison with Zhang et al.'s protocol [47] and Tu et al.'s protocol [48] are summarized in Table 2. As can be seen, the computation cost of our improved protocol is same as Tu et al.'s protocol.

7 Conclusions

In this paper, we analyzed Tu et al.'s password-based authenticated key agreement protocol. We pointed out that Tu et al.'s protocol suffers from impersonation attack by which an attacker can masquerade as a legal server to share common session keys with legal users. Moreover, we proposed an improvement of Tu et al.'s protocol to overcome the security problem. The security of the improved protocol was proved in the random oracle model. Our improvements did not change the computational and communication cost of the original protocol.

References

1. Rosenberg J et al (2002) SIP: session initiation protocol. IETF, rfc 3261
2. Li JS, Kao CK, Tzeng JJ (2011) VoIP secure session assistance and call monitoring via building security gateway. Int J Commun Syst. doi:10.1002/dac.1191
3. Chen WE, Huang YL, Lin YB (2010) An effective IPv4-IPv6 translation mechanism for SIP applications in next generation networks. Int J Commun Syst. doi:10.1002/dac.1040
4. Chen WE, Lin PJ (2010) A performance study for IPv4-IPv6 translation in IP multimedia core network subsystem. Int J Commun Syst. doi:10.1002/dac.1071
5. Chiu KL, Chen YS, Hwang RH (2011) Seamless session mobility scheme in heterogeneous wireless networks. Int J Commun Syst. doi:10.1002/dac.1189
6. Cho K, Pack S, Kwon TT, Choi Y (2010) An extensible and ubiquitous RFID management framework over next-generation networks. Int J Commun Syst. doi:10.1002/dac.1073
7. Chiang WK, Chang WY (2010) Mobile-initiated network-executed SIP-based handover in IMS over heterogeneous accesses. Int J Commun Syst. doi:10.1002/dac.1115
8. Chen MX, Wang FJ (2010) Session integration service over multiple devices. Int J Commun Syst. doi:10.1002/dac.1109
9. Geneiatakis D, Dagiuklas T, Kambourakis G, Lambrinoudakis C, Gritzalis S, Ehlert S (2006) Survey of security vulnerabilities in session initiation protocol. IEEE Commun Surv Tutorials 8(3):68–81
10. Farash MS, Attari MA (2014) An efficient and provably secure three-party password-based authenticated key exchange protocol based on Chebyshev chaotic maps. Nonlinear Dyn 77(1–2):399–411
11. Farash MS, Bayat M, Attari MA (2011) Vulnerability of two multiple-key agreement protocols. Comput Electr Eng 37(2):199–204
12. Farash MS, Attari MA (2012) An id-based key agreement protocol based on ECC among users of separate networks. In: 9th international ISC conference on information security and cryptology (ISCISC'12), pp 32–37
13. Farash MS, Attari MA (2014) A pairing-free ID-based key agreement protocol with different PKGs. Int J Netw Secur 16(2):143–148
14. Farash MS, Attari MA (2014) An enhanced and secure three-party password-based authenticated key exchange protocol without using server's public-keys and symmetric cryptosystems. Inf Technol Control 43(2):143–150
15. Farash MS, Attari MA (2014) Cryptanalysis and improvement of a chaotic maps-based key agreement protocol using Chebyshev sequence membership testing. Nonlinear Dyn 76(2):1203–1213
16. Bayat M, Farash MS, Movahed A (2010) A novel secure bilinear pairing based remote user authentication scheme with smart card. In: IEEE/IFIP international conference on embedded and ubiquitous computing (EUC), pp 578-582
17. Farash MS, Attari MA, Atani RE, Jami M (2013) A new efficient authenticated multiple-key exchange protocol from bilinear pairings. Comput Electr Eng 39(2):530–541
18. Farash MS, Attari MA (2013) Provably secure and efficient identity-based key agreement protocol for independent PKGs using ECC. ISC Int J Inf Secur 5(1):18–43
19. Farash MS, Attari MA, Bayat M (2012) A certificateless multiple-key agreement protocol without hash functions based on bilinear pairings. Int J Eng Technol 4(3):321–325
20. Farash MS (2014), Cryptanalysis and improvement of an efficient mutual authentication RFID scheme based on elliptic curve cryptography. J Supercomput. doi:10.1007/s11227-014-1272-0
21. Farash MS, Attari MA (2014) An anonymous and untraceable password-based authentication scheme for session initiation protocol using smart cards. Int J Commun Syst. doi:10.1002/dac.2848
22. Farash MS, Attari MA (2014) An improved password-based authentication scheme for session initiation protocol using smart cards without verification table. Int J Commun Syst. doi:10.1002/dac.2879
23. Farash MS, Attari MA (2014) An efficient client-client password-based authentication scheme with provable security. J Supercomput. doi:10.1007/s11227-014-1273-z
24. Farash MS, Attari MA (2014) A secure and efficient identity-based authenticated key exchange protocol for mobile client-server networks. J Supercomput 69(1):395–411
25. Sadat Mousavi-nik S, Yaghmaee-moghaddam MH, Ghaznavi-ghoushchi MB (2012) Proposed secureSIP authentication scheme based on elliptic curve cryptography. Int J Comput Appl 58(8):25–30
26. Yoon E, Yoo K, Kim C, Hong Y, Jo M, Chen H (2010) A Secure and efficient SIP authentication scheme for converged VoIP networks. Comput Commun 33(14):1674–1681
27. Wang F, Zhang Y (2008) A new provably secure authentication and key agreement mechanism for SIP using certificateless public-key cryptography. Comput Commun 31:2142–2149
28. Dimitris G, Costas L (2007) A lightweight protection mechanism against signaling attacks in a SIP-Based VoIP environment. Telecommun Syst 36(4):153–159
29. Wu L, Zhang Y, Wang F (2009) A new provably secure authentication and key agreement protocol for SIP using ECC. Comput Standards Interfaces 31(2):286–291
30. Liao Y, Wang S (2010) A new secure password authenticated key agreement scheme for SIP using self-certified public keys on elliptic curves. Comput Commun 33(3):372–380
31. Wu S, Pu Q, Kang F (2013) Practical authentication scheme for SIP. Peer-to-Peer Network Appl 6(1):61–74
32. He D, Chen J, Chen Y (2012) A secure mutual authentication scheme for session initiation protocol using elliptic curve cryptography. Secur Comm Netw 5:1423–1429
33. Yang CC, Wang RC, Liu WT (2005) Secure authentication scheme for session initiation protocol. Comput Secur 24:381–386

34. Huang HF, Wei WC, Brown GE (2006) A new efficient authentication scheme for session initiation protocol. In: 9th joint conference on information sciences
35. Jo H, Lee Y, Kim M, Kim S, Yang'sOff-line password-guessing attack to Yang's and Huang's authentication schemes for session initiation protocol. In: Fifth international joint conference on INC IMSandIDC pp 618–621 (2009)
36. Durlanik A, Sogukpinar I (2005) SIP authentication scheme using ECDH. World Enformatika Socity Transations Eng Comput Technol 8:350–353
37. Yoon EJ (2009) Yoo KY. Cryptanalysis of DS-SIP authentication scheme using ECDH. In: 2009 international conference on new trends in information and service science, pp 642–647
38. Liu FW, Koenig H (2011) Cryptanalysis of a SIP authentication scheme. In: communications and multimedia security. Springer, Berlin/Heidelberg, pp 134–143
39. Tsai JL (2009) Efficient nonce-based authentication scheme for session initiation protocol. Int J Netw Secur 8(3):312–316
40. Yoon EJ, Yoo KY (2009) A new authentication scheme for session initiation protocol. In: 2009 international conference on complex, intelligent and software intensive systems, CISIS, pp 549–554
41. Chen TH, Yeh HL, Liu PC, Hsiang HC, Shih WK (2010) A secured authentication protocol for SIP using elliptic curves cryptography. CN, CCIS 119:46–55
42. Arshad R, Ikram N (2011) Elliptic curve cryptography based mutual authentication scheme for session initiation protocol. Multimed Tool Appl. doi:10.1007/s11042-011-0787-0
43. Tang H, Liu X (2012) Cryptanalysis of Arshad et al.'s ECC-based mutual authentication scheme for session initiation protocol. Multimed Tool Appl. doi:10.1007/s11042-012-1001-8
44. Yoon E, Shin Y, Jeon I, Yoo K (2010) Robust mutual authentication with a key agreement scheme for the session initiation protocol. IETE Techn Rev 27(3):203–213
45. Xie Q (2012) A new authenticated key agreement for session initiation protocol. Int J Commun Syst. doi:10.1002/dac.1286
46. Farash MS, Attari MA (2013) An enhanced authenticated key agreement for session initiation protocol. Inf Technol Control 42(4):333–342
47. Zhang L, Tang S, Cai Z (2013) Efficient and flexible password authenticated key agreement for voice over internet protocol session initiation protocol using smart card. Int J Commun Syst. doi:10.1002/dac.2499
48. Tu H, Kumar N, Chilamkurti N, Rho S (2014) An improved authentication protocol for session initiation protocol using smart card. Peer-to-Peer Netw Appl. doi:10.1007/s12083-014-0248-4
49. Abdalla M, Pointcheval D (2005) Interactive Diffie-Hellman assumptions with applications to password-based authentication. In: Proceedings of FC'05, LNCS 3570, pp 341–356

Mohammad Sabzinejad Farash received the B.Sc. degree in Electronic Engineering from Shahid Chamran College of Kerman in 2006, and the M.Sc. degree in Communication Engineering from I. Hussein University in 2009. He also received the Ph.D. degree in Cryptographic Mathematics at the Department of Mathematics and Computer Sciences of Kharazmi University in Iran in 2013. His research interests are Security Protocols and Provable Security Models.