Multimed Tools Appl (2013) 66:165–178
DOI 10.1007/s11042-011-0787-0

Elliptic curve cryptography based mutual authentication scheme for session initiation protocol

R. Arshad · N. Ikram

Published online: 13 April 2011

Abstract The Session Initiation Protocol (SIP) is the most widely used signaling protocol for controlling communication on the internet, establishing, maintaining, and terminating the sessions. The services that are enabled by SIP are equally applicable in the world of multimedia communication. Recently, Tsai proposed an efficient nonce-based authentication scheme for SIP. In this paper, we do a cryptanalysis of Tsai's scheme and show that Tsai's scheme is vulnerable to the password guessing attack and stolen-verifier attack. Furthermore, Tsai's scheme does not provide known-key secrecy and perfect forward secrecy. We also propose a novel and secure mutual authentication scheme based on elliptic curve discrete logarithm problem for SIP which is immune to the presented attacks.

Keywords Authentication · Elliptic curve cryptosystem · Security · Session initiation protocol

1 Introduction

The Session Initiation Protocol (SIP) [7] is one of the most important protocols that support multimedia services on both wired as well as wireless networks. SIP is an application layer control (signalling) protocol that can create, modify and terminate sessions with one or more participants. These sessions include multimedia distribution, internet telephone calls and internet multimedia conferences. Session members can communicate via a mesh of unicast relations or via multicast or a combination of these. Authentication is the most important aspect for SIP. When a user wants to access a SIP server in order to get various services from the remote server, he/she has to perform an

R. Arshad (✉) · N. Ikram
National University of Sciences and Technology, Rawalpindi, Pakistan
e-mail: raziarshad@hotmail.com

N. Ikram
e-mail: nassar@nust.edu.pk

authentication process. Therefore, the security of SIP is becoming more important which calls for robust authentication scheme. The original authentication scheme for SIP does not provide strong security, as it is derived from HTTP digest authentication [5] and has been proven to be insecure [19].

Recently, various authentication schemes were proposed to provide strong security for SIP [1, 5, 6, 13, 14, 17, 19]. In the basic authentication scheme adopted in SIP, the SIP server uses challenge-response mechanism to verify the identity of the user only. In 2005, Yang et al. [19] pointed out that the procedure of the original SIP authentication scheme based on HTTP digest authentication is vulnerable to offline password guessing attack and server spoofing attack. In order to overcome these flaws, Yang et al. proposed a secure authentication scheme for SIP. This authentication scheme is based on Diffie-Hellmann Key Exchange [3], which depends on the difficulty of discrete logarithms. The computational cost of Yang et al.'s authentication scheme is very high, making it unsuitable for platforms offering low computational power. In order to improve this limitation, Durlanik et al. [4] proposed an authentication scheme using Elliptic Curve Diffie-Hellmann (ECDH) key exchange algorithm in 2005. Later in 2009, Tsai [16] proposed an efficient nonce-based authentication scheme. Since all the communication messages are encrypted/decrypted by using one-way hash function and XOR operation, its computation cost is low, making it promising for low-power processors.

Notwithstanding suitability of Tsai's scheme for low power processors, Tsai's scheme is still vulnerable to offline password guessing attack and stolen-verifier attack while it does not provide known-key secrecy and perfect forward secrecy [10]. This paper demonstrates the vulnerability of Tsai's authentication scheme to the above attacks, and then proposes a secure and efficient authentication based on the Elliptic Curve Discrete Logarithm Problem (ECDLP) for SIP in order to overcome such security flaws.

The rest of the paper is organized as follows. Section 2 introduces the SIP architecture, SIP authentication procedure, elliptic curve cryptography and secure one-way hash function. Section 3 gives the review of the Tsai's authentication scheme. Section 4 discusses the cryptanalysis of Tsai's authentication scheme. In Section 5, an efficient and secure mutual authentication scheme of SIP is proposed. We analyse the security and efficiency of our proposed scheme in Sections 6 and 7. Finally, we conclude the paper in Section 8.

2 Preliminaries

2.1 SIP architecture

SIP, using client-server architecture over HTTP, is based on uniform resource locator and uniform resource identifier. Text-encoding scheme and the header format of SIP are the same as proposed in SMTP, i.e., SIP reuses SMTP headers such as To, From, Date, and subject [4]. The proxy server, redirect server, user agent, register server, and location server are the main components of SIP architecture [19]. The function of each component is described as follows.

- **Proxy server:** A proxy server always forwards a request and response between a callee and a caller. When the proxy server receives a request, it forwards the request to the current location of the callee, and then forwards the response from the callee to the caller.

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

2.2 SIP authentication procedure

The security of SIP authentication is based on the challenge-response mechanism [19]. Before the authentication procedure starts, the client user preshares a password with the server. This preshared password is used to verify the identity of the client user or the server. The original SIP authentication scheme proceeds as follows. See Fig. 1.

Step 1 Client→Server: REQUEST
The client sends a REQUEST to the server.

Step 2 Server→Client: CHALLENGE (nonce, realm)
The server generates a CHALLENGE that includes a nonce and the client's realm. Note that the realm is used to prompt the username and password. Then the server sends a CHALLENGE back.

Step 3 Client→Server: RESPONSE (nonce, realm, username, response)
The client computes a response = F (nonce, username, password, realm). Note that F (.) is a one-way hash function and is used to generate a digest authentication message. Then, the client sends the RESPONSE to the server.

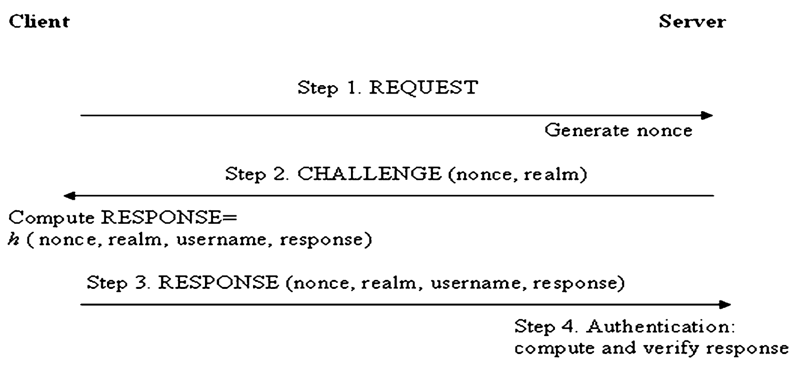
Client
Server
Step 1. REQUEST
Generate nonce
Step 2. CHALLENGE (nonce, realm)
Compute RESPONSE=
h (nonce, realm, username, response)
Step 3. RESPONSE (nonce, realm, username, response)
Step 4. Authentication:
compute and verify response

Fig. 1 SIP authentication procedure

Step 4 According to the username, the server extracts the client's password. Then the server verifies whether the nonce is correct or not. If it is correct, the server computes F (nonce, username, password, realm) and uses it to compare with the response. If they match, the server authenticates the identity of the client.

2.3 Elliptic curve cryptography

Let $p>3$be a large prime number. An elliptic curve E over F_P is the set of all solutions $(x, y) \in F_P * F_P$ to an equation $E : y^2 = x^3 + ax + b \bmod p$, where a, $b \in F_P$ and $4a^3 + 27b^2 \neq 0 \bmod p$, together with a special point O called the point at infinity. It is well known that E is an (additively written) abelian group with the point O serving as its identity element. Choose a generator point $P = (x_P, y_P)$ over $E(F_P)$, where $P \neq O$. In such a way, a subgroup G of the elliptic curve group$E(F_P)$ is constructed. To guard against small subgroup attacks, the point P should be of a prime order $n$or equivalently, $nP = O$ and we should have $n > 4\sqrt{p}$. To protect against other known attacks on special classes of elliptic curves [15], n should not divide$p^i–1$ for all $1 \leq i \leq V$(V=20 suffices in practice), $n \neq p$ should be satisfied, and the curve should be non-super singular. To retain the intractability of ECDLP, n should at least satisfy $n>2^{160}$.Let us consider three related mathematical problems in G; the Elliptic Curve Discrete Logarithm Problem (ECDLP), the Elliptic Curve Computational Diffie-Hellman Problem (ECCDHP) and the Elliptic Curve Decisional Diffie-Hellman Problem (ECDDHP).

- **ECDLP**: Given a point element Q in G, find an integer $x \in Z_q^*$such that $Q = xP$, where xP indicates that the point P is added to itself for x times by the elliptic curves operation.
- **ECCDHP**: For $a, b \in Z_q^*$, given any two point elements aP, bP in G, compute abP in G.
- **ECDDHP**: For $a, b, c \in Z_q^*$, given any three point elements aP, bP and cP in G, decide whether $cP = abP$.

Clearly, we have the relationship that the ECCDHP is no harder than ECDLP, and ECDDHP is also no harder than ECCDHP in G. Therefore, we assume that ECDDHP is intractable ($E : y^2 = x^3 + ax + b \bmod p$ is a non-super singular elliptic curve), which may guarantee that there is no polynomial time algorithm to solve ECDDHP, ECCDHP and ECDLP with non negligible probability.

2.4 One-way hash function

A one-way hash function H is said to be secure, if it satisfies the following properties [2].

- Given x, it is easy to compute $H(x) = y$. However, when given y, it is hard to compute $H^{-1}(y) = x$.
- Given x, it is computationally infeasible to find $x' \neq x$ such that $H(x') = H(x)$.

3 Review of Tsai's authentication scheme for SIP

In this section, we briefly review Tsai's nonce-based authentication scheme for SIP [16]. There are two phases in Tsai's scheme: registration and authentication. Notations used in this paper are defined in Table 1

Table 1 Notations

U	The client user
S	The server
D	A uniformly distributed dictionary of size $\|D\|=2^k$, where $40\leq k\leq 105$
PW	A low-entropy password of U extracted from D
K_S	A high-entropy secret key of S, which is only known by the server and must be safeguarded.
SK	A shared common session key between U and S
G,P	Subgroup of the elliptic curve group $E(F_P)$ and its generator point of order q.
h	Secure one-way hash function, where $h:\{0,1\}^* \to \{0,1\}^l$ and $l=160$
$\oplus$	A bit-wise XOR operation
$\|\|$	A concatenation operation

3.1 Registration phase

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

Step 1 $U \to S$: {username, PW}
U submits his/her username and PW to S.

Step 2 S stores U's username and PW in the user account database.

3.2 Authentication phase

If a legal user U wants to login into the system, he/she must type his/her username and PW. All steps of authentication phase are then executed as follows.

Step 1 $U \to S$: REQUEST (username, N_C)
U generates a random number N_C and then sends it with a request message as REQUEST (username, N_C) to S.

Step 2 $S \to U$: CHALLENGE (realm, $N_S \oplus h(PW\|N_C), h(PW\|N_S\|N_C)$)
When S receives the request message, S generates a random nonce N_S. Then, S uses N_S, N_C and PW to compute $N_S \oplus h(PW\|N_C)$ and $h(PW\|N_S\|N_C)$.Finally, S sends a challenge message CHALLENGE (realm, $N_S \oplus h(PW\|N_C), h(PW\|N_S\|N_C)$) to U.

Step 3 $U \to S$: RESPONSE (username, realm,h(username, realm, $h(N_S\|PW\|N_C)$))
When U receives the challenge message, U uses N_C, PW to compute $h(PW\|N_C)$ and derivesN_S by computing $N_S \oplus h(PW\|N_C) \oplus h(PW\|N_C)$. Then, U computes $h(PW\|N_S\|N_C)$ and verifies whether it is equal to the received challenge $h(PW\|N_S\|N_C)$.If the two are not equal, U rejects the server challenge message. Otherwise, U authenticates$S$ and computes two hash values $h(N_S\|PW\|N_C)$ and h(username, realm, $h(N_S\|PW\|N_C)$). Finally, U sends a response message RESPONSE (username, realm, h(username, realm, $h(N_S\|PW\|N_C)$)) to S.

Step 4 When S receives the response message, S uses N_S, N_C, PW to compute $h(N_S\|PW\|N_C)$. If the computed $h(N_S\|PW\|N_C)$ is not the same as response $h(N_S\|PW\|N_C)$, then$S$ rejects the user request. Otherwise,S accepts the connection.

After mutual authentication between S and U, $SK = N_S$ is used as a session key. See Fig. 2.

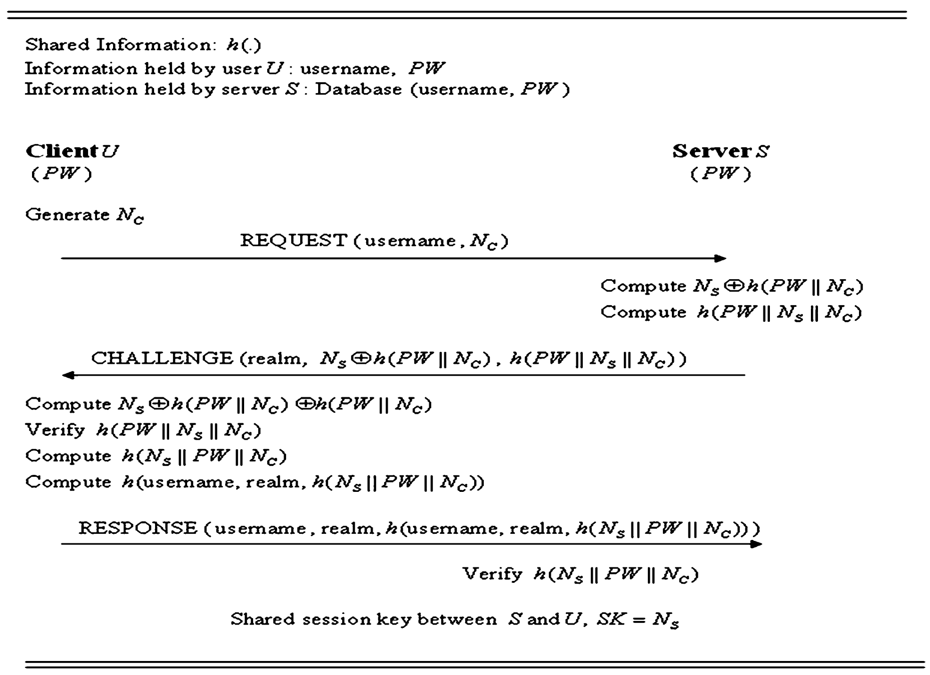
Shared Information: $h(.)$
Information held by user U: username, PW
Information held by server S: Database (username, PW)
Client U (PW)
Server S (PW)
Generate N_C
REQUEST (username, N_C)
Compute $N_S \oplus h(PW \| N_C)$
Compute $h(PW \| N_S \| N_C)$
CHALLENGE (realm, $N_S \oplus h(PW \| N_C)$, $h(PW \| N_S \| N_C)$)
Compute $N_S \oplus h(PW \| N_C) \oplus h(PW \| N_C)$
Verify $h(PW \| N_S \| N_C)$
Compute $h(N_S \| PW \| N_C)$
Compute h(username, realm, $h(N_S \| PW \| N_C)$)
RESPONSE (username, realm, h(username, realm, $h(N_S \| PW \| N_C)$))
Verify $h(N_S \| PW \| N_C)$
Shared session key between S and U, $SK = N_S$

Fig. 2 Tsai's authentication scheme for session initiation protocol

4 Cryptanalysis of Tsai's authentication scheme for SIP

In this section, we show that Tsai's authentication scheme is vulnerable to password guessing attack, stolen-verifier attack and does not provide perfect forward secrecy and known-key secrecy.

4.1 Password guessing attack

In password guessing attack, an adversary intercepts authentication messages and stores them locally and then attempts to use a guessed password to verify the correctness of his/her guessing using these authentication messages[9, 11]. In Tsai's authentication scheme, an adversary can intercept N_C,$N_S \oplus h(PW\|N_C)$ and $h(PW\|N_S\|N_C)$. Then, the adversary can guess a password PW^* from dictionary$D$and computes $h(PW^*\|N_C)$. He/she derives N_S^* by computing $N_S^* = N_S \oplus h(PW\|N_C) \oplus h(PW^*\|N_C)$.Then, he/she computes $h(PW^*\|N_S^*\|N_C)$. If it is equal to $h(PW\|N_S\|N_C)$, then the guess password PW^* is correct. Otherwise, the adversary repeatedly guesses a new password PW^*. Thus, Tsai's authentication scheme is vulnerable to password guessing attacks.

4.2 Stolen-verifier attack

In most existing password based authentication schemes, servers are always the prime targets of the adversaries, because the users' verifiers (e.g. passwords) are stored in server's database. In stolen-verifier attack [21], an adversary who steals a password-verifier from the server can use it directly to impersonate a legitimate user in a user authentication execution. In fact, an adversary who obtains password-verifier may further mount a guessing attack.

In Tsai's authentication scheme, the plaintext password PW of the user is stored in the server, can be eavesdropped and then used to masquerade as the original user. Tsai did not explain the stolen-verifier attack, with regard to obtaining the user's secret data PW, which is stored in the server. By obtaining user's secret data PW, an illegitimate user can login to the server as a legitimate user. Suppose an adversary has stolen the user's secret data PW from the server. Then, adversary can choose a random nonce N_E and send it with a request message as REQUEST (username, N_E) to S in step 1 of authentication phase. Then, S will send a challenge message CHALLENGE (realm, $N_S \oplus h(PW\|N_E), h(PW\|N_S\|N_E)$) to adversary. Then, adversary can make a valid response message RESPONSE (username, realm, h(username, realm, $h(N_S\|PW\|N_E)$)) and send it as a response message to S. As a result, S will authenticate adversary as a legal user U and accepts the user's login request. Therefore, the adversary can easily impersonate the legal user U. In addition, adversary can also impersonate the legal server by using the stolen user's secret password PW. Therefore, Tsai's authentication scheme is insecure against stolen-verifier attack.

4.3 Known-key secrecy

The known-key secrecy means the compromise of a past session key can't derive any other session keys or long term private key (e.g., U's password PW or S private key K_S). The compromise of session key enables the protocol to be compromised.

In Tsai's authentication scheme, suppose an adversary has a session key $SK = N_S$ of the protocol. Then, by using $SK = N_S$, the adversary computes $h(PW\|N_C)$because of $N_S \oplus h(PW\|N_C) \oplus SK$. Since N_C is the open nonce value and PW included in $h(PW\|N_C)$ is also known to adversary by performing password guessing attack. That is, adversary makes a guess at the secret password PW^* from the dictionary D and checks if $h(PW\|N_S)? = h(PW^*\|N_C)$. If it holds, the adversary has guessed the correct password $PW^* = PW$. Compromise of user password PW will enable the adversary to impersonate S or U freely. Therefore, Tsai's authentication scheme cannot achieve known-key secrecy.

4.4 Perfect forward secrecy

The important security requirement for strong protocol evaluation is the perfect forward secrecy. A protocol with perfect forward secrecy ensures that even if one entity's long-term key is compromised, it will never reveal any session keys used previously [18, 20]. It is easily seen that Tsai's authentication scheme cannot achieve the perfect forward secrecy. When a user's password PW is compromised, all the session keys $SK = N_S$ will also be opened and hence previous communication messages will be learnt.

In Tsai's authentication scheme, suppose an adversary obtains the password PW from the compromised user and N_C is also known because adversary can intercept it from the public channel. After that, adversary can compute the session key $SK = N_S$ from $N_S \oplus h(PW\|N_C)$ because of the $N_S \oplus h(PW\|N_C) \oplus h(PW\|N_C)$. Thus, if we know PW and N_C then any old short term session keys are compromised. Therefore, Tsai's authentication scheme cannot achieve the perfect forward secrecy.

5 Proposed authentication scheme for SIP

In this section, we propose our novel mutual authentication scheme for session initiation protocol. The proposed scheme will consist of two phases: the registration phase and the authentication phase. Description of each phase is as follows:

5.1 Registration phase

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

Step 1 $U \rightarrow S$: {username, PW}
U submit his/her username and PW to S. S computes two secret values HPW and HK_S by using hash of U's username, PW and K_S. Here $HPW = h$(username$||$ PW) and $HK_S = h$(username$||K_S$).

Step 2 S computes the password verifier $VPW = HPW \oplus HK_S$ for U.

Step 3 S stores U's username and VPW in the user account database.

5.2 Authentication phase

If a legal user wants to login into S, he/she must type his/her username and PW. All steps of authentication phase executed as follows.

Step 1 $U \rightarrow S$: REQUEST (username, R_1)
U generates a random integer $r_1 \in Z_q^*$. U computes $HPW = h$(username$||PW$) from username and PW respectively. Then, U computes $R_1 = (HPW.r_1)P$ and sends it with a request message as REQUEST (username, R_1) to$S$.

Step 2 $S \rightarrow U$: CHALLENGE (realm, R_2, h_1)
Upon receiving the request message, S extracts HPW from VPW by computing $HK_S = h$(username$||K_S$) and $HPW = VPW \oplus HK_S$, where VPW is a stored password verifier for U in the user account database. Then, S computes $R_1' = HPW^{-1}R_1 = (HPW^{-1}.HPW.r_1)P = r_1P$. Here, HPW^{-1} is computed by using Extended Euclidean Algorithm over Z_q^*. Now, S generates another random integer $r_2 \in Z_q^*$ and computes $R_2 = r_2P, SK_S = r_2R_1' = r_2r_1P = r_1r_2P$ and $h_1 = h(SK_S||R_2)$, where SK_S is the secret session key and $SK_S = r_2r_1P = r_1r_2P$ holds due to commutative property of elliptic curve group. Finally, S sends a challenge message CHALLENGE (realm, R_2, h_1) to U.

Step 3 $U \rightarrow S$: RESPONSE (username, realm, h(username$||$realm$||SK_U$))
Upon receiving the challenge message, U computes a secret session key $SK_U = r_1R_2 = r_1r_2P$ and checks that $h(SK_U||R_2)? = h_1$. If these are not equal, U rejects the server challenge message. Otherwise, U authenticates S and computes a message authentication code h(username$||$realm$||SK_U$). Finally, U sends a response message RESPONSE (username, realm, h(username$||$realm$||SK_U$)) to$S$.

Step 4 Upon receiving the response message, S computes h(username$||$realm$||SK_S$) and verifies whether it is equal to the received response h(username$||$realm$||SK_U$). If they are not equal, S rejects the user response message. Otherwise, S authenticates U and accepts the user's login request.

After mutual authentication between U and S, $SK = SK_U = SK_S = r_1r_2P$ is used as a shared session key. See Fig. 3.

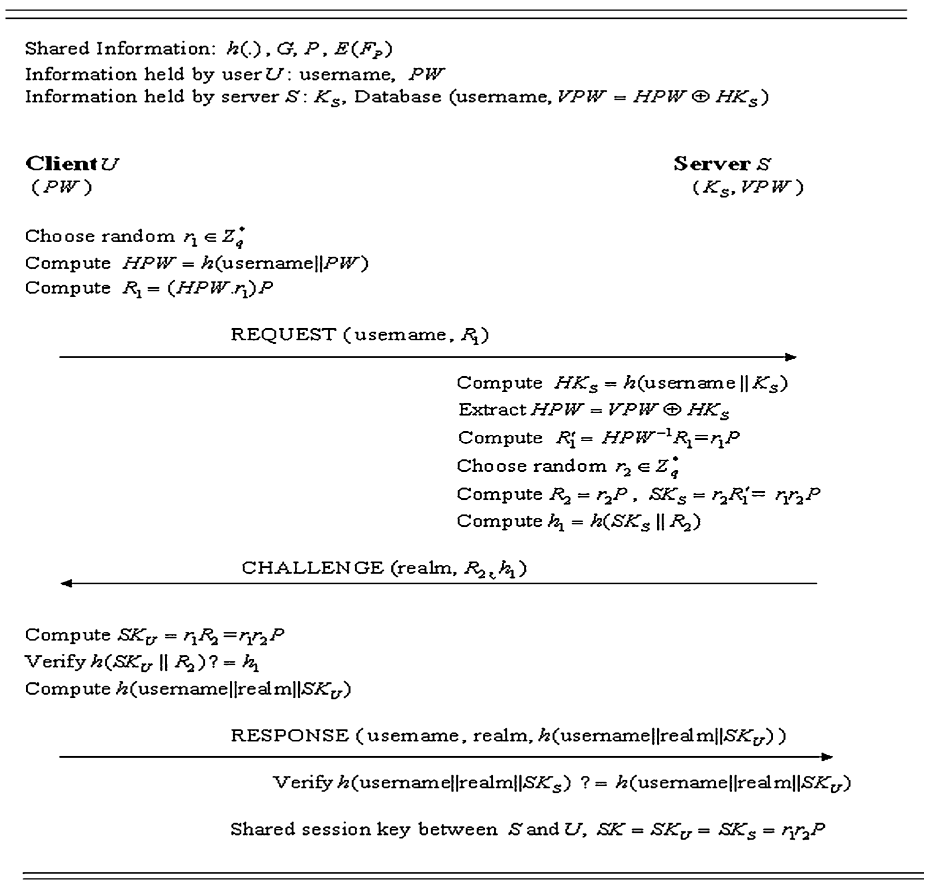
Shared Information: $h(.)$, G, P, $E(F_p)$
Information held by user U: username, PW
Information held by server S: K_S, Database (username, $VPW = HPW \oplus HK_S$)
Client U (PW)
Server S (K_S, VPW)
Choose random $r_1 \in Z_q^*$
Compute $HPW = h(\text{username}||PW)$
Compute $R_1 = (HPW.r_1)P$
REQUEST (username, R_1)
Compute $HK_S = h(\text{username}||K_S)$
Extract $HPW = VPW \oplus HK_S$
Compute $R_1' = HPW^{-1}R_1 = r_1P$
Choose random $r_2 \in Z_q^*$
Compute $R_2 = r_2P$, $SK_S = r_2R_1' = r_1r_2P$
Compute $h_1 = h(SK_S || R_2)$
CHALLENGE (realm, R_2, h_1)
Compute $SK_U = r_1R_2 = r_1r_2P$
Verify $h(SK_U || R_2) ?= h_1$
Compute $h(\text{username}||\text{realm}||SK_U)$
RESPONSE (username, realm, $h(\text{username}||\text{realm}||SK_U)$)
Verify $h(\text{username}||\text{realm}||SK_S) ?= h(\text{username}||\text{realm}||SK_U)$
Shared session key between S and U, $SK = SK_U = SK_S = r_1r_2P$

Fig. 3 Proposed authentication scheme for session initiation protocol

6 Security analysis

In this section, we examine the security of our proposed authentication scheme in terms of the following security properties: replay attack, password guessing attack, modification attack, stolen-verifier attack, server spoofing attack, man-in-middle attack, session key security, known-key secrecy and perfect forward secrecy.

6.1 Replay attack

In our proposed scheme, nonce variables r_1 and r_2 are generated independently and both will be different in each login message. Suppose, an adversary intercepts REQUEST (username, R_1)in Step 1 of authentication phase and replays it to impersonate U to login into S. However, adversary cannot compute a correct session key SK and deliver it to S in step 3 of authentication phase unless he/she correctly guesses password PW to obtain r_1P and guesses the right r_2 from $R_2 = r_2P$. When an adversary tries to guess r_1 from r_1P and r_2 from $R_2 = r_2P$, he/she will face ECDLP. On the other hand, suppose an adversary intercepts CHALLENGE (realm, R_2,h_1) from S in step 2 of the authentication phase and replays it to impersonate S. For the same reason, if an adversary cannot gain the correct r_1 from r_1P, U will find out that h_1 is not equivalent to his/her computed $h(SK_U||R_2)$. Then, U will not send RESPONSE (username, realm, h(username||realm||SK_U)) back to adversary in step 3 of authentication phase. Therefore, our proposed authentication scheme can resist the replay attack.

6.2 Password guessing attack

An online password guessing attack cannot succeed since S can choose appropriate trial intervals. On the other hand, in an offline password guessing attack, there must be no verification information for passwords in all exchanges. Observe our proposed scheme, if an adversary obtains all the exchanged messages (R_1, R_2, h_1, h(username, realm, SK_U)) by passive attack, and wants to guess U's password, he first guesses a password $PW^*$and uses it to compute $R_1' = h(\text{username}||PW^*)^{-1}R_1 = r_1P$. By using R_1' and R_2, the adversary will try to compute the session key $SK = SK_U = SK_S = r_1r_2P$. However, adversary has to break the ECCDHP or ECDDHP to find the keying material $SK = r_1r_2P$ from $R_1' = r_1P$ and $R_2 = r_2P$ to verify his/her guess. But, the adversary cannot gain the session key without $r1$ of $r_1.P$ and r_2 of $r_2.P$. Therefore, our proposed scheme can resist password guessing attack.

6.3 Modification attack

An adversary may modify the communication messages (R_1, R_2, h_1, h(username||realm|| SK_U)) being transmitted over an insecure network. However, although adversary forges them, our proposed scheme can detect this modification attack, because it can verify not only the equality of the session key $SK = SK_U = SK_S = r_1r_2P$ computed by each party, but also the correctness of R_1 and R_2 transmitted between two parties by validating the message authentication code $h_1 = h(SK_S||R_2)$, h(username||realm||SK_U) in the proposed scheme. Therefore, our proposed scheme can resist the modification attack.

6.4 Stolen-verifier attack

Servers are always the prime target of attacks. An adversary may acquire $VPW = HPW \oplus HK_S$ stored in S. However, without knowing K_S, adversary cannot forge a login request to pass the authentication, as HPW is hidden in $VPW = HPW \oplus HK_S$ by using server's secret key K_S. Thus, the correctness of the guessed password HPW^* cannot be verified by checking $HPW^* = HPW$. Therefore, our proposed scheme can resist stolen-verifier attack.

6.5 Server spoofing attack

Since our proposed scheme provides mutual authentication, the server spoofing attack [12] can be resisted. In the authentication phase, as U sends REQUEST (username, R_1) to the adversary masquerading as the server, the adversary cannot generate proper (R_2,h_1) without knowing K_S in Step 2 of the authentication phase. Therefore, the server spoofing attack doesn't work in our proposed scheme.

6.6 Man-in-middle attack

In our proposed scheme, an adversary cannot pretend to be U or S to authenticate the other since he/she doesn't know the password PW. Suppose adversary has recorded one of the

U's previous authentication message, say REQUEST (username, R_1) and RESPONSE (username, realm, h(username||realm||SK_U)). If adversary knows the password PW corresponding to the intercepted R_1, he/she can generate a random number $i \in Z_q^*$ to compute $R_1^{'} = (h(\text{username}||PW).i)P$ and send the forged authentication message REQUEST (username, $R_1^{'}$) to S. Next, adversary can intercept the message CHALLENGE (realm, R_2, h_1) sent back by S to U, where $h_1 = h(SK_S||R_2)$ and $SK_S = r_2R_1^{'} = r_2iP = ir_2P$, compute $SK_U^{'} = iR_2$, and then forge the authentication message $h(\text{username}||\text{realm}||SK_U^{'})$ to fool S. However, since adversary does not know HPW and r_1, he/she cannot compute $SK_U = r_1R_2 = r_1r_2P$ to impersonate U to fool S in this manner. The forged challenge message CHALLENGE (realm, R_2, h_1) to fool U can be also detected by U because adversary does not know HPW to extract $r_1$fromR_1 = ($HPW.r_1$)P as same reason. Therefore, our proposed scheme can resist the man-in-middle attack.

6.7 Session key security

The session key $SK = SK_U = SK_S = r_1r_2P$ is only known to U and S. The random values r_1 and r_2 are protected by the ECDLP and ECCDHP and the secure one-way hash function. Therefore, our proposed scheme provides session key security.

6.8 Known-key secrecy

Although, an adversary obtains the fresh session key $SK = SK_U = SK_S = r_1r_2P$, he/she cannot obtains U's secret password PW from $R_1 = (h(\text{username}||PW).r_1)P$ by using password guessing attack because he/she will face ECDLP to obtain r_1 from r_1P. If the adversary obtains r_1P, then he/she can guess PW from $R_1 = (h(\text{username}||PW).r_1)P$ by computing $R_1^{'} = (h(\text{username}||PW)^{-1}.h(\text{username}||PW).r_1)P? = R_1$. However, it is infeasible because U never sends the r_1P in the plaintext form over the open network. It is always encrypted by using the secret password PW such as $R_1 = (h(\text{username}||PW).r_1)P$. Also, adversary is unable to compute two values r_1P and r_2P from the fresh session key $SK = SK_U = SK_S = r_1r_2P$ due to hardness of ECDLP and ECCDHP. Therefore, our proposed authentication scheme provides the feature of known-key secrecy.

6.9 Perfect forward secrecy

Perfect forward secrecy is provided in the situation that even though U's secret password PW or S's secret key K_S is compromised; an adversary still cannot obtain any previous session keys. In our proposed authentication scheme, suppose that an adversary knows U's secret password PW or S's secret key K_S, he tries to determine the session key SK for past sessions and decrypt them. Since adversary is still faced with the hardness of ECDLP and ECCDHP to compute the session key $SK = SK_U = SK_S = r_1r_2P$ from the two extracted values r_1P and r_2P. Additionally, suppose an adversary may acquire $VPW = HPW \oplus HK_S$ stored in S. By knowing S's secret key K_S, he/she extracts $HPW^* = VPW \oplus h(\text{username}||K_S)$.The adversary intercept all the exchange messages (R_1, R_2, h_1, h (username, realm, SK_U)) by passive attack. Now, he/she can compute $R_1^{'} = HPW^{*-1}R_1 = ((HPW^{*-1}.HPW).r_1)P = r_1P$. Since adversary is still faced with the hardness of ECDLP to compute the value r_1 from r_1P. Therefore, our proposed authentication scheme can provide the property of perfect forward secrecy.

The security properties of previous related schemes [16, 19] and the proposed authentication scheme are summarized in Table 2.

Table 2 Comparison of security properties

	Yang et al.'s Scheme	Tsai's Scheme	Proposed Scheme
Replay attack	Secure	Secure	Secure
Password guessing attack	Secure	Insecure	Secure
Modification attack	Secure	Secure	Secure
Stolen-verifier attack	Insecure	Insecure	Secure
Mutual Authentication	Provided	Provided	Provided
Session key security	Not Applicable	Provided	Provided
Known key secrecy	Not Applicable	Not Provided	Provided
Perfect forward secrecy	Not Applicable	Not Provided	Provided

7 Performance comparison

The computational costs of our proposed authentication scheme and the previous related schemes [16, 19] are summarized in Table 3. Since, our proposed authentication scheme is based on the elliptic curve cryptosystem [8], the total overhead for communication and performance can be reduced. For example, to reach a reasonable security level, it just requires a 160-bit prime p to construct the elliptic curve group $E(F_P)$.

The proposed authentication scheme requires five ECC multiplications during the protocol. These five ECC computations are needed to provide known-key secrecy and perfect forward secrecy. The cost of inversion operation in Z_q^* is negligible. When considering Tsai's scheme, our proposed authentication scheme is though computationally a little intensive, but offer enhanced security. However, effects of computational intensity can easily be mitigated by the use of powerful processors.

8 Conclusion

In this paper, we have shown that Tsai's authentication scheme for session initiation protocol is vulnerable to password guessing attack and stolen verifier attack. Furthermore, it does not provide known-key secrecy and perfect forward secrecy. In order to resolve these security problems, we have proposed a novel and secure mutual authentication scheme based on the elliptic curve discrete logarithm problem for session initiation protocol. The proposed authentication scheme not only resists these attacks but also provides greater security and efficiency. Hence, our proposed authentication scheme can be executed faster than any other previously proposed related schemes.

Table 3 Comparison of computational cost

	Yang et al.'s Scheme	Tsai's Scheme	Proposed Scheme
No. of Exponentiation	4	0	0
No. of ECC Computation	0	0	5
No. of hash function	8	7	8
No. of exclusive OR	4	3	2
Security	DLP	Hash	ECDLP

References

1. Arkko J, Torvinen V, Camarillo G, Niemi A, Haukka T (2002) Security mechanism agreement for SIP sessions. IETF Internet Draft (draft-ietf-sip-sec-agree-04.txt)
2. Damgard I (1989) A design principle for hash functions. Advances in Cryptology, CRYPTO'89, LNCS 1989, (435): 416–427
3. Diffie W, Hellman M (1976) New directions in cryptology. IEEE Transaction on Information Theory 22(6)
4. Durlanik A, Sogukpinar I (2005) SIP Authentication Scheme using ECDH. World Enformatika Socity Trans Eng Comput Technol 8:350–353
5. Franks J, Hallam-Baker P, Hostetler J, Lawrence S, Leach P, Luotonen A (1999) HTTP authentication: basic and digest access authentication. IETF RFC2617
6. 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
7. Handley M, Schulzrinne H, Schooler E, Rosenberg J (1999) SIP: session initiation protocol. IETF RFC2543
8. Koblitz N (1987) Elliptic curve cryptosystems. Math Comput 48:417–426
9. Lee CC (2009) On security of an efficient nonce based authentication scheme for SIP. Int J Netw Secur 9 (3):201–203
10. Lin CL, Hwang T (2003) A password authentication scheme with secure password updating. Comput Secur 22(1):68–72
11. Lu R, Cao Z (2006) Off-line password guessing attack on an efficient key agreement protocol for secure authentication. Int J Netw Secur 3(1):35–38
12. Lu R, Cao Z (2008) A simple user authentication scheme for grid computing. Int J Netw Secur 7(2):202–206
13. Rosenberg J, Schulzrinne H, Camarillo G, Johnstone A, Peterson J, Sparks R (2002) SIP: session initiation protocol. IETF RFC3261
14. Thomas M (2001) SIP security requirements. IETF Internet Draft (draftthomas-sip-sec-reg-00. txt) (work in progress)
15. Toorani M, Shirazi AAB (2009) A directly public verifiable signcryption scheme based on elliptic curves. Proceedings of the 14th IEEE Symposium on Computers and Communications (ISCC 09), pp. 713–716
16. Tsai JL (2009) Efficient nonce-based authentication scheme for session initiation protocol. Int J Netw Secur 8(3):312–316
17. Veltri L, Salsano S, Papalilo D (2002) SIP security issues: the SIP authentication procedure and its processing load. IEEE Netw 16(6):38–44
18. Wang B, Li ZQ (2006) A forward-secure user authentication scheme with smart cards. Int J Netw Secur 3(2):116–119
19. Yang CC, Wang RC, Liu WT (2005) Secure authentication scheme for session initiation protocol. Comput Secur 24:381–386
20. Yoon EJ, Yoo KY (2009) A new authentication scheme for session initiation protocol. 3rd International Workshop on Intelligent, Mobile and Internet Services in Ubiquitous Computing (IMIS 2009), pp. 549–544
21. Yoon EJ, Shin YN, Jeon IS, Yoo KY (2010) Robust mutual authentication with a key agreement scheme for the session initiation protocol. IETE Tech Rev 27(3):203–213

R. Arshad received his M.Sc. degree in Mathematics from Quaid-i-Azam University, Islamabad, Pakistan in 2001; the M.Sc. degree in Computer Science from International Islamic University, Islamabad, Pakistan in 2004; the M.S. degree in Information Security from Sichuan University, Chengdu, China in 2007. He is currently research associate in National University of Sciences and Technology (NUST), Pakistan. His current research interest includes multimedia system security and Cryptography.

N. Ikram graduated in Electrical Engineering from NED, Karachi, Pakistan in 1987. He received his M.Sc. in Military Electronics and Systems Engineering from Royal Military College of Sciences, Shrivenham, Cranfield University, UK in 1995 and PhD in information Security from Bradford University, UK in 1999. He is currently Adjunct Professor at National University of Sciences and Technology, Islamabad, Pakistan. His current research interests include multimedia system security and cryptography.