SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks (2013)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.834

RESEARCH ARTICLE

A secure authentication scheme for session initiation protocol by using ECC on the basis of the Tang and Liu scheme

Azeem Irshad*, Muhammad Sher, Muhammad Shahzad Faisal, Anwer Ghani, Mahmood Ul Hassan and Shehzad Ashraf Ch

Department of Computer Science & Software Engineering, International Islamic University, Islamabad, Pakistan

ABSTRACT

Session initiation protocol (SIP) provides the basis for establishing the voice over internet protocol sessions after authentication and exchanging signaling messages. SIP is one of the significant and extensively used protocols in the multimedia protocol stack. Since the RFC2617 was put forth, numerous schemes for SIP authentication have been presented to overcome the flaws. Recently, in 2012, Tang and Liu proposed SIP based authentication protocol and claimed for eliminating the threats in Arshad and Ikram protocol. However the scheme can be made more robust by making further improvements, as the former scheme may come under a threat by adversaries through impersonating a server, given that the user password is compromised. We have proposed an improved protocol for SIP authentication by using elliptic curve cryptography that encounters the previous threat with enhanced security. The analysis shows that proposed scheme is suitable for applications with higher security requirements. Copyright © 2013 John Wiley & Sons, Ltd.

KEYWORDS

session initiation protocol; elliptic curve cryptography; authentication; voice over internet protocol; security

***Correspondence**

A. Irshad, IIU, International Islamic University, Islamabad, Pakistan.
E-mail: azeem.phdcs66@iiu.edu.pk

1. INTRODUCTION

Session initiation protocol (SIP) is developed by the Internet Engineering Task Force and is responsible for establishing, maintaining, and terminating sessions [1,3]. It works at application layer and is a text-based client–server protocol. The SIP supports internet telephony calls and provides multimedia-based services. The multimedia communication sessions are based on voice and video calls that are supported over internet protocols such as hyper text transfer protocol (HTTP) or simple mail transfer protocol [2]. SIP was adopted by the 3rd Generation Partnership Project (3GPP) as the protocol for session establishment of multimedia-based applications in 3G mobile networks. The voice over internet protocol-based networks provides the convergence of networks, whereas the SIP-based next generation networks support web-based control of multimedia services.

Authentication [22–26] is one of the most significant aspects in SIP. By using SIP, the client puts a request to the server for establishing a voice call session on the basis of a few shared or globally known parameters. The client needs to verify that it is connected with the SIP user agent of server, instead of an attacker. The authentication can be performed in several ways for different applications such as password-based authentication as a one-time password (challenge-response password or password list) [28], public-key cryptography, zero-knowledge proofs, digital signatures, and other authentication protocols such as secure socket layer (SSL) [30], IP security [31], secure shell [32], and Kerberos [33]. These authentication procedures depend upon different applications and the computing resources. The SIP mutual authentication is based on the combination of password-based authentication and public key cryptography, as the former does not need much processing power.

Numerous authentication schemes have been proposed to date [4,5,10–16,18,25–36] with different vulnerabilities [17]. The original authentication scheme, HTTP digest authentication based on RFC2617 [2], fails to provide enough security. Afterwards, Yang, Wang, and Liu [4] in 2005 proposed a protocol for SIP authentication, but that was vulnerable to offline password-guessing attack and server spoofing attack. This scheme was based on the Diffie–Hellman key exchange algorithm supported by the difficulty of Discrete Logarithm Problem (DLP). Durlanik in 2005 proposed an efficient SIP authentication scheme by using elliptic curve cryptography (ECC) [6–9]. However,

Durlanik's scheme [5] was vulnerable to Denning–Sacco attacks and stolen verifier attacks. Wu, Zhang, and Whang in 2009 [10] provides an increased level of security for ECC-based protocol by using the Canetti–Krawczyk security model. Yoon et al. [11] in 2010 proved an offline password guessing attack in the Wu, Zhang, and Whang scheme and presented an efficient scheme for converged voice over internet protocol networks by using ECC. Pu [12] found a password guessing attack in Yoon et al. Tsai [13] in 2009, also proposed a protocol on the basis of one-way hash function and exclusive-OR (XOR) function. Arshad and Ikram [14] found known-key secrecy, perfect forward secrecy, password guessing attack, and stolen verifier attack in the Tsai scheme and presented an improved protocol to counter the previous threats. Debiao, Jianhua, and Yitao [27] presented an efficient scheme, yet it might expose a few shared parameters on the compromise of server secret. The Tang and Liu [15] scheme identified an offline password guessing attack in the Arshad and Ikram scheme and presented an efficient protocol for SIP authentication.

In existing the scheme we review the Tang and Liu scheme and show that this protocol is conditionally vulnerable to misrepresentation threat by an adversary. The Tang and Liu scheme is quite robust to the extent of password guessing attack or session key secrecy; however, in the presence of numerous clients for a single server, the chances for a password theft of a client are always there. The user, ignorant of the fact that its password has been compromised, might establish a session with an attacker holding its own password misrepresenting as a server. In this way, an attacker can misrepresent the user as a server entirely eliminating the server entity. The objective of this paper is to improve the Tang and Liu scheme to eliminate the threat posed to the user as identified in the paper. We have proposed a new protocol to counter this threat with improved security features and in an almost equivalent computation cost.

The rest of the paper is organized as follows. In Section 2, the procedure for SIP authentication and background for ECC has been described. Section 3 reviews the Tang and Liu scheme [15] and present cryptanalysis with drawbacks. Section 4 describes the proposed scheme of the SIP authentication protocol. Section 5 presents the security analysis, and Section 6 illustrates performance analysis with comparison, whereas the last section concludes the findings.

2. PRELIMINARIES

In this section, the SIP architecture [11] and the background for ECC [8,37] have been described.

2.1. Session initiation protocol architecture

The SIP is a text-oriented request-response protocol. The SIP works much alike the HTTP principle. In the internet, we use uniform resource locator to recognize resources such as websites, whereas in SIP-based authentication, uniform resource identifier [11] is used to identify users, particularly their phone numbers or names (e.g., SIP: user1@iiu.cs.pk). The SIP architecture comprises a few components such as the user agent client, the proxy server, the redirect server, the register server, and the location server. A user agent signifies the terminal. The proxy server acts as an intermediary party on the behalf of client and server. A redirect server informs the caller about the location of the callee, so that the caller may contact the callee directly. A register server lets the caller update its most recent location in the location server, whereas a register server lets the caller update its most recent location in the location server.

2.2. Session initiation protocol authentication procedure

A client registers with the proxy server to become a member. In registration [11], the client exchanges some secret parameters such as the password on a secure channel with the server. The client authenticates with the proxy server and tries to login on the basis of pre-shared secret each time it establishes a new session. After getting authenticated with the proxy server, the SIP session procedure is performed to locate another client/user through proxy server to establish a session. The SIP authentication protocol includes the following messages.

(1) Client → server: REQUEST

The client sends a REQUEST to the server, initially.

(2) Server → client: CHALLENGE (nonce and realm)

After receiving the REQUEST, the server sends a CHALLENGE to the client with a nonce and realm. The realm prompts the user to input username and password.

(3) Client → server: RESPONSE (nonce, realm, username, and response)

The client generates a RESPONSE as h (nonce, realm, username, and response), whereas h() is a one-way hash function and is sent to the server.

(4) The server receives the RESPONSE message, calculates the user's password, and verifies the nonce authentication after the computation of h(nonce, realm, username, and response). If it is not verified, the server terminates the session. Otherwise, it proceeds to acknowledge the client in positive. A unique and mutual session key is generated on both sides after the successful authentication procedure.

2.3. Elliptic curve cryptography

This subsection accommodates some of the basic ECC concepts pertinent to this paper. The ECC [6–8] security

has been proven to be more efficient cryptographic scheme as compared with earlier conventional techniques [9]. This technique provides an equivalent level of security with much less key sizes. The mathematical operations are defined over an EC equation Ep(a, b): $y^2 = x^3 + ax + b(\text{mod } p)$ and $4a^3 + 27b^3 \neq 0$ (mod p), where a, b ∈ Fp, and 'p' is a large prime number. Both values a, b defines the EC, whereas the points (x, y) that satisfy the former statement including a point at infinity lies on the EC. The scalar multiplication is performed using vP = P + P + ,…,Pv given a point P and an integer v ∈ F^*p. All domain parameters such as (p, a, b, G, n and h) belong to the finite field, F^*p. E is an abelian group and the point at infinity serves as identity element for this group.

3. REVIEW AND CRYPTANALYSIS FOR THE TANG AND LIU SCHEME

In this section, the review and cryptanalysis based on the Tang and Liu scheme has been described.

3.1. Brief review for the Tang and Liu scheme

This subsection reviews the Tang and Liu scheme [15] that has been found vulnerable to an impersonating attack.

The Table I lists some notations that are used in the forthcoming sections.

The Tang and Liu scheme contains four phases: the system setup phase, the registration phase, the authentication phase, and the password change phase.

(1) System setup phase

In this phase, all the entities agree on globally known EC parameters. The server publishes all of these *p, a, b, P, n, h*, and *Q* except the secret key, K_s.

(2) Registration phase

Table I. Notations

U_i, U	ith user or user
ID_i, username	username or user-id
S	server username or id
PW_i:	low entropy password of user
K_s	high entropy password of server
SK	a session key
$P_{pub} = K_s . P$	server public key
h(.)	cryptographic one-way hash operation
H(.)	a function making a point map to another point on elliptic curve
A	an adversary or attacker
⊕	exclusive-OR (XOR)
‖	concatenation operation
→	a common channel
C ? = B	checks, whether C equals to B
D	a uniformly distributed dictionary of size |D|

In the registration phase, the user communicates with the server over a secure channel, and the messages exchanged in this phase are mentioned later.

Step I The user selects its identity ID_i and password PW_i, and sends them over some secure channel such as the virtual private network or SSL.

Step II The server computes $VPW_i = h(ID_i \| K_s) \oplus PW_i$ and stores (ID_i, VPW_i) in the database.

(3) Login and authentication phase

Figure 1 shows the login and authentication phase between the user and the server. The authentication procedure is performed on a common channel. The details are given later.

Step I: $\underline{U_i} \rightarrow S$: REQUEST($ID_i$, R_u); *U* chooses a random number $r_1 \in Z_n^*$, computes $R = r_1.P$, $R_u = R + H(ID_i \| PW_i)$, and sends this request message REQUEST(ID_i, R_u) to the server.

Step II: $S \rightarrow U_i$: CHALLENGE(S, R_s, h_1): *S* checks, whether the *IDi* exists in its database. If it does not match, the session is aborted, otherwise, S computes the $PW_i = VPW_i \oplus h(ID_i \| K_s)$, $R' = R_u - H(ID_i, PW_i)$. Afterwards, S chooses a random number $r_2 \in Z_n^*$ and computes $R_s = r_2.P$, $SK_s = r_2.R'$, $h_1 = (S \| ID_i \| R' \| R_s \| Sk_s)$.Finally, it sends a message CHALLENGE(S, R_s, h_1) to U.

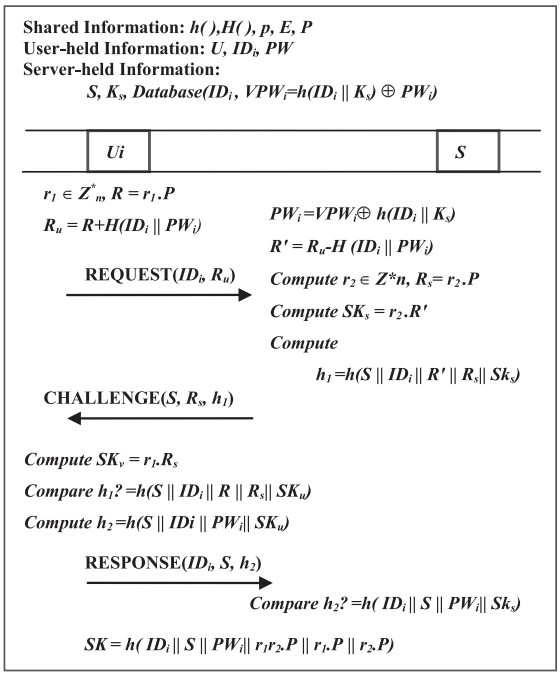

Figure 1. Login and authentication phase of the Tang and Liu scheme

Step III: $U_i \rightarrow S$: RESPONSE(ID_i, S, h_2); U computes $SK_u = r_1\ R_s$ and checks whether h_1 is equal to $h(S \| ID_i \| R \| R_s \| SK_u)$; if it does not match, U terminates the session, otherwise computes $h_2 = (S \| ID_i \| PW_i \| Sk_u)$ and sends the response RESPONSE(ID_i, S, h_2) to S.

Step IV: Upon receiving the message, S verifies whether h_2 equates $h(ID_i \| S \| PW_i \| SK_s)$. If it matches, S proceeds with the establishment of an agreed session key SK with the user, that is, $SK = h(ID_i \| S \| PW_i \| r_1r_2.P \| r_1.P \| r_2.P)$.

(4) Password change phase

When the user feels that his or her password is stolen, he or she can modify the password by establishing a new session with the server on the basis of existing password over the common channel. The messages exchanged for changing the password are shown later:

Step I: The user performs the login and authentication phase for creating the usual session before changing the password. After successful session establishment, the user inputs a new password PW_i^* to replace the old one.

Step II: $Ui \rightarrow S$: T, h_1; the user computes $T = h(SK \| SK_u)\ PW_i^*$ and $h_1 = (SK \| SK_u \| PW_i^*)$, and sends T, h_1 to the server.

Step III: Upon receiving the message, the server computes $PW_i^* = h(SK \| SK_s)$ and then checks whether h_1 is equal to $h(SK \| SK_s \| PW_i^*)$. If it does not match, it terminates the session and sends the message "Denied", h_3 for $h_3 = h(SK \| SK_s \| \text{"Denied"})$ and otherwise sends "Accepted", h_4 for $h_4 = h(SK \| SK_s \| \text{"Accepted"})$. Finally, the server computes $VPW_i^* = h(ID_i \| K_s) \oplus PW_i^*$ and replaces VPW_i by VPW_i^*.

3.2. Attacks on the Tang and Liu scheme

The Tang and Liu scheme is vulnerable to a misrepresentation attack given that the password of a user is compromised. The attack can be launched using the following procedure.

Step I: $U_i \rightarrow A$: REQUEST(IDi, R_u); U chooses a random number, $r_1 \in Z^*_n$, $R = r_1.P$ and computes $R_u = R + H(ID_i, \| PW_i)$ and sends the request message REQUEST(ID_i, R_u) to S. Adversary A intercepts the message and maneuver with U to impersonate S.

Step II: $A \rightarrow U_i$: CHALLENGE(S, R_A, h_1); A, having the password need not compute PW_i through V_i, so it directly computes the $R' = R_u - H(ID \| PW_i)$. Afterwards, A chooses a random number $r_2 \in Z^*_n$ and computes $R_A = r_2.P$, $SK_A = r_2.R'$, $h_1 = (S \| ID_i \| R' \| R_A \| SK_A)$. Finally, it sends a message CHALLENGE(S, R_A, h_1) to U.

Step III: $U_i \rightarrow A$: RESPONSE(ID_i, S, h_2); U computes $SK_u = r_1R_A$ and checks whether h_1 is equal to $h(S \| ID_i \| R \| R_A \| Sk_u)$; if it does not match, U terminates the session, otherwise computes $h_2 = (S \| ID_i \| PW_i \| SK_u)$ and sends the response RESPONSE(ID_i, S, h_2) to A.

Step IV: On the receipt of message, A verifies whether h_2 equates $h(ID_i \| S \| PW_i \| SK_A)$ and A proceeds with the generation of an agreed session key SK with the user, that is, $SK = h(ID_i \| S \| PW_i \| r_1r_2.P \| r_1.P \| r_2.P)$

3.3. Drawbacks on the Tang and Liu scheme

An adversary A needs to compromise only the password of U_i, and can easily masquerade the identity of server towards U_i without approaching the server-held parameters such as password verifier VPW_i and secret K_s. In this way, the U_i can be made to believe that it is connected with the server, although it might be connected with an adversary. An attacker can impersonate the identity of S with its (server's) absence in totality, during the session establishment with U. The Tang and Liu protocol did not employ the server's identity (K_s) appropriately in the challenge message generated by S towards U. The adversary maneuvers U with the mere knowledge of its (U's) own password, impersonating S, giving U the impression of a legitimate session establishment with the server.

4. PROPOSED AUTHENTICATION MODEL

We propose a new ECC-based SIP authentication scheme to overcome the flaws in the Tang and Liu scheme. This authentication scheme contains three phases: the system setup phase, the registration phase, and the authentication phase.

4.1. System setup phase

In this phase, all the entities agree on globally known EC parameters. The server publishes all of these p, a, b, P, n, h, and P_{pub} except the secret key, K_s.

4.2. Registration phase

In the registration phase, the user, U_i communicates with the server over a secure channel, and the parameters exchanged in this phase are mentioned later.

Step I: The user selects its identity ID_i and password PW_i, and sends over a secure channel such as virtual private network or SSL.

Step II: On the receipt of parameters, the server computes $V_i = h(ID_i \| K_s) \oplus PW_i$ and stores (ID_i, V_i) in its database.

4.3. Authentication phase

The authentication phase is responsible for establishing a unique session key among the entities on the basis of pre-shared parameters in registration phase. The shared parameters are $h(.)$, $H(.)$, p, E, P, and P_{pub} [27]. Figure 2 shows the authentication phase between the user and the server. The authentication procedure is performed on a common channel. The details for the phase are given later.

Step I $U_i \rightarrow S$: REQUEST(ID_i, R_1); U_i chooses a random number $r_1 \in Z^*_n$, computes $R = r_1.P$, $R_1 = R + H(ID_i \| PW_i)$, and sends this request message REQUEST(ID_i, R_1) to the server.

Step II $S \rightarrow U_i$: CHALLENGE(S, R_2, h_1): S checks whether the ID_i exists in its database. If it does not match, the session is aborted, otherwise, S computes the $PW_i = V_i \oplus h(ID_i \| K_s)$, $R' = R_u - H(ID_i \| PW_i)$. Next, S chooses a random number $r_2 \in Z^*_n$ and computes $R_2 = r_2.P$, $R_1'' = K_s \times R'$, $h_1 = (S \| ID_i \| R_1 \| R_1'' \| R_2)$. Finally, it sends a message CHALLENGE(S, R_2, h_1) to U.

Step III $U_i \rightarrow S$: RESPONSE(ID_i, $Realm$, h_2); U checks whether h_1 is equal to $h(ID_i \| R_1 \| R_1' \| R_2)$; if it does not match, U terminates the session, otherwise U computes $SK_u = r_1.R_2$ and $h_2 = h(ID_i \| Realm \| R_1 \| R_1' \| R_2 \| SK_u \| PW_i)$ and sends the response RESPONSE(ID_i, $Realm$, h_2) to S.

Step IV On the receipt of message, the S verifies whether h_2 equates $h(ID_i \| Realm \| R_1 \| R_1'' \| R_2 \| SK_s \| PW_i)$. If it matches, the S proceeds with the establishment of an agreed session key SK with the user and computes $SK_s = r_2.R'$ and sends a positive acknowledgement to user. In this manner, the entities establish an agreed and unique session key $SK = SK_u = SK_s = r_1 r_2.P = r_1.R_2 = r_2.R'$.

The public key P_{pub} has been published and shared in the same manner like other parameters. If the user tries the wrong public key of server erroneously without verification, it would not be able to meet the challenge presented from server and h_1 will not match $h(ID_i \| R_1 \| R_1' \| R_2)$ in Figure 2.

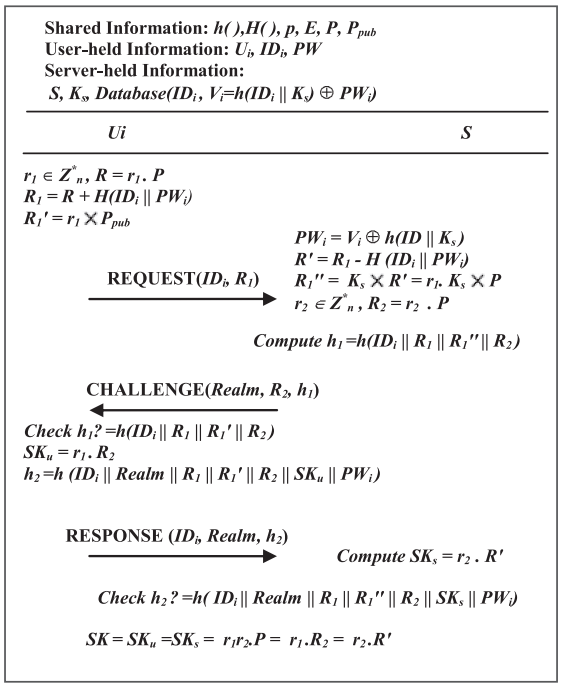

Figure 2. Authentication phase of proposed protocol

5. SECURITY ANALYSIS

First, we describe some of the security terms needed for security analysis.

Term 1: An EC computational DHP (ECCDHP) is stated as follows: given a G's generator P, $a \times P$, $b \times P$ to compute $ab \times P$.

Term 2: A one-way hash operation as $y = h(x)$, where it is a hard problem to compute x, given y, in the previous equation.

Term 3: The ECDLP is stated as follows: given a point $Q = aP$ on EC, it would be hard enough to compute a, given Q and P.

5.1. Dening–Sacco attack

The Dening–Sacco attack is activated when an attacker tries to guess either a user's password or the server's long term secret key or another session key out of an old compromised session key.

The proposed scheme resists *Dening–Sacco attack*.

Proof. If an old session key SK gets compromised, it requires R_1' or R_1'' in $h_2 = h(ID_i \| Realm \| R_1 \| R_1' \| R_2 \| SK_u \| PW_i)$ for guessing the right password PW on an offline basis. Alternatively, it needs to face ECCDHP to break $SK_u = r_1.r_2.P$.

5.2. Stolen verifier attack

The attacker can steal valuable information from the server; if it maintains the user's information such as passwords in its database and use it to impersonate the legitimate users for its own cause that is known as stolen verifier attack.

The proposed scheme resists *stolen verifier attack*.

Proof. If an attacker A steals the verifier V_i from server database, he cannot guess the right password until it also obtains the secret K_s of the server by using stealth.

5.3. Replay attack

The replay attacks are launched when an adversary replays the genuine message parameters at some other time to deceive or impersonate any legitimate participant.

The proposed scheme resists *replay attack*.

Proof. An adversary A cannot reuse information that is obtained from an open network. A may intercept the message REQUEST(ID_i, R_1) and replay sometime to the server. The server sends the message CHALLENGE ($Realm$, R_2, h_1) to A. A cannot generate h_2 because it needs R_1' parameter. Hence, the replay attack cannot be possible.

5.4. Password guessing attack

In password guessing attack, an adversary tries to guess the secret parameters such as the user password or the long term secret of the server, by applying brute-force attack, out of intercepted messages that may be based on some function of original secret [19,20].

The proposed scheme resists the *password guessing attack*.

Proof. There are two messages REQUEST(ID_i, R_1) and RESPONSE(ID_i, $Realm$, h_2) including h_2 and R_1, generated as the function of the password along with other parameters. Adversary A needs to guess R that is a function mapped point to an EC point, which is a hard problem. A needs to obtain both R_1' and SK_u or R_1" and SK_s, for guessing the password from h_2. If, however, A succeeds in compromising the session key, it might not access R_1' or R_1". Hence, the scheme is resistant to password guessing attack.

5.5. Perfect forward secrecy

The perfect forward secrecy suggests maintaining the secrecy of previous session keys, if the long-term private keys of an entity, that is, either a user or a server is compromised.

The proposed scheme provides *perfect forward secrecy* [21].

Proof. The perfect forward secrecy means that if long-term private keys of one or more entities are compromised, the secrecy of old session keys will remain intact. To compromise an old session key, the adversary A needs to break $SK_u = r_1.\ R_2$, that faces CDHP. If A compromises the user password PW, it cannot generate the current session key by masquerading as a server, without compromising the secret K_s. On the other hand, if A steals the secret K_s, it needs the password verifier Vi, to access the user's PW. Hence, A cannot spoof the opposite entity's identity, if one of the PW or K_s has been compromised. The identity of user, however, depends only on PW, once it is compromised, the attacker can generate a session key until it is not updated by the valid user.

5.6. Impersonation attack

An impersonation attack may also be referred as masquerading or spoofing attack. In this attack, an adversary impersonates the identity of a legitimate user to some other user in an illegal manner.

The proposed scheme can resist an *impersonation attack* in a scenario where the password of the user has been compromised, that would have been possible in the Tang and Liu scheme because the server does not use K_s for calculating any parameter such as R_s, h_1, and R', besides retrieving U's PW_i.

Proof. An adversary A may generate a random number $r_1 \in Z_n^*$ and compute $R = r_1.P$ and $R_1' = r_1 \times P_{pub}$. However, A cannot generate R_1 because it needs PW to take a point map function with R. However, it might assume any random R_A and send a request REQUEST(ID_i, R_A) to server. The server may compute an R_1" on the basis of received value R_A and send it as $h_1 = h(ID_i \| R_1 \| R_1" \| R_2)$. A can neither compute a valid session key SK_u nor generate h_2 and would fail the test for $h_2 = ?\ h(ID_i \| Realm \| R_1 \| R_1' \| R_2 \| SK_u \| PW_i)$ at the server.

5.7. Modification attack

The modification attacks can be launched if an adversary modifies and reconstructs the message contents in an authorized manner to present it to any legitimate user.

The proposed scheme resists *modification attack*, because all the parameters exchanged (ID_i, R_1, R_2, h_1 and h_2) during the protocol can be verified from h_1 and h_2.

5.8. Mutual authentication

The mutual authentication defines that both entities authenticate each other in the same authentication protocol.

The proposed scheme provides *mutual authentication*, because either of the two entities verifies the other entity on account of password PW or secret Ks and the exchanged hash values h_1 and h_2.

5.9. Man-in-the-middle attack

This attack is launched by an adversary to act as a silent intermediary between the intended participants and make them believe that these are talking to each other but as a matter of fact the participants would be talking to adversary if the attack is successful.

The proposed scheme resists *man-in-the-middle attack*, because the proposed scheme provides mutual authentication and the possibility of man-in-the-middle attack is eliminated.

5.10. Session key security

The session key security signifies the knowledge of the established session key to only the legitimate participants, that is, the user and the server, and nobody else.

The proposed scheme provides *session key security*, which means after creating a session the session key *SK* is only known to the concerned entities. The random values r_1 and r_2 cannot be guessed out of R_1 and R_2 or other hash values because of ECDLP, ECCDHP, and one-way hash function.

5.11. Known-key security

The known-key security defines the concept of generation of a unique session key between the two legal participants for each run of authentication protocol.

The proposed scheme provides *known-key security*, because the user always generates $R = r_1.P$ by creating new random integers r_1 each time a session is created and sends them to the server. The server computes a scalar multiplication operation of received value R and a self-generated random integer r_2 and creates a session key. The server in return, generates the session key by using the same procedure and sends $R_2 = r_2.P$ toward the user, which is scalar multiplied with random integer r_1. The use of random integers necessitates the uniqueness of the established session key each time a session is created.

6. PERFORMANCE ANALYSIS AND COMPARISON

In this section, the performance for the Tang and Liu scheme has been evaluated and compared with the proposed scheme. The total computation cost for this scheme is shown in Table II. For better understanding, we define the notations, each representing the computation cost of a single operation in terms of time [27], thereby, T_{Smul} for EC scalar multiplication operation, T_{Mmul} for a modular multiplication operation, T_h for one-way hash operation, T_{XOR} for a string exclusive-OR operation, T_{H-P} for a hash-to-point operation and T_{PA} for a point addition operation.

In the proposed scheme, the security has been enhanced with an added cost. Although, a few operations are more costly than others, such as T_{Smul}, being the scalar multiplication operation, takes more computation than other ones; therefore, the tendency must be to reduce the number of T_{Smul} operations in the construction of protocol to a level such that the security is not compromised. So, this is now considered as more significant than other operations for comparing the overhead cost of different authentication protocols. Nevertheless, there is a tradeoff between security and cost optimization [38,39]. With the increase in security, the cost also increases and vice versa. We want less cost of computations in the protocol but not on the cost of security or a possible attack. If a protocol comes under any kind of attack, one can question its

Table II. Comparison between Tang and Liu and our protocol

	Schemes	
Costs	Tang and Liu scheme	Proposed scheme
Computational cost (client)	$2T_{Smul} + 3T_h + 1T_{H-P} + 1T_{PA}$	$3T_{Smul} + 2T_h + 1T_{H-P} + 1T_{PA}$
Computational cost (server)	$2T_{Smul} + 4T_h + 1T_{H-P} + 1T_{PA} + 1T_{XOR}$	$3T_{Smul} + 3T_h + 1T_{XOR} + 1T_{H-P} + 1T_{PA}$
Total	$4T_{Smul} + 7T_h + 2T_{H-P} + 2T_{PA} + 1T_{XOR}$	$6T_{Smul} + 5T_h + 1T_{XOR} + 2T_{H-P} + 2T_{PA}$

Table III. Comparison of security features

	Schemes				
Attacks/security features	Durlanik scheme	Tsai scheme	Yoon et al. scheme	Tang and Liu scheme	Proposed scheme
Impersonation attack					
Replay attack	S	S	S	S	S
Password guessing attack	IS	IS	IS	S	S
Modification attack	S	N/A	S	S	S
Stolen verifier attack	N/A	IS	IS	S	S
Mutual Authentication	SP	SP	SP	SP	SP
Session key security	SP	SP	SP	SP	SP
Known-key secrecy	SP	SnP	SP	SP	SP
Perfect forward secrecy	SP	SnP	SP	SP	SP

N/A, not applicable; S, secure; IS, insecure; SP, security provided; SnP, security not provided.

performance despite being computationally efficient. The Tang and Liu scheme does not contain any function of its secret key in the challenge message that renders the user to a conditional misrepresentation attack.

The Tang and Liu scheme comprises $4T_{Smul} + 7T_h + 2T_{H-P} + 2T_{PA} + 1T_{XOR}$, whereas the proposed scheme contains $6T_{Smul} + 5T_h + 1T_{XOR} + 2T_{H-P} + 2T_{PA}$ messages as a whole. The Tang and Liu scheme incurs four scalar multiplication computations (T_{Smul}) and seven hash operations (T_h) in a single run of protocol. On the other hand, the proposed protocol incurs six T_{Smul} and five T_h operations, that is, the proposed protocol incurs two additional T_{Smul} operations in comparison with the Tang and Liu scheme to reinforce its defense against identified threats. Alternatively, the Tang and Liu scheme incurs less cost on the compromise of security.

We do not claim that the proposed scheme is efficient, rather it is more secure than Tang and Liu scheme. Hence, the proposed scheme encounters the misrepresentation threat on the cost of two additional scalar multiplications. The proposed scheme provides a higher level of security for hostile environments with some additional cost as compared with the Tang and Liu scheme.

In Table III, we have made a comparison of security properties of different authentication protocols. The Tang and Liu scheme has been shown as insecure only in the case of impersonation attack, whereas the proposed scheme has been secure to all identified attacks in the table.

7. CONCLUSIONS

In this scheme, we have shown that the Tang and Liu protocol for providing SIP-based authentication is vulnerable to misrepresentation attack, given that the user password has been compromised. To overcome the shortcomings, a new robust protocol has been proposed that not only eliminates the threats posed to the user as identified in the Tang and Liu scheme but also provides an improved security with almost equivalent number of messages or computation costs. Hence, the scheme has the potential for being deployed in situations with improved security requirements.

ACKNOWLEDGEMENTS AND FUTURE WORK

The authors would like to thank the anonymous reviewers for their productive comments that helped us to improve the quality of this paper. In the future, the authors intend to work on further cost optimization, along with finding the alternative ways of using a password verifier table that is maintained by the server, to authenticate the user, or at least minimizing dependence on verifier table.

REFERENCES

1. Rosenberg J, Schulzrinne H, Camarillo G, Johnston A, Peterson J, Sparks R, Handley M, Schooler E. IP: session initiation protocol, IETF RFC3261 June 2002.
2. Franks J, Hallam-Baker P, Hostetler J, Lawrence S, Leach P, Luotonen A, Stewart L. HTTP authentication: basic and digest access authentication', IETF RFC2617 June 1999.
3. Thomas M. SIP Security Requirements. IETF Internet Draft Nov2001.
4. Yang CC, Wang RC, Liu WT. Secure authentication scheme for session initiation protocol. *Computers and Security* 2005; **24**(5):381–386.
5. Durlanik A, Sogukpinar I. SIP authentication scheme using ECDH, World Enformatika Society Transaction on Engineering Computing and Technology, 2005; 350–353.
6. Certicom Research Standard for efficient cryptography, Elliptic Curve Cryptography Version. 1.0. SEC 1 2000.
7. Koblitz N. Elliptic curve cryptosystems. *Mathematics of Computation* 1987; **48**:203–209.
8. Miller V. Uses of elliptic curves in cryptography. In: *Advances in Cryptology CRYPTO'85, LNCS 218*, Springer-Verlag, Berlin, 1986; 417–426.
9. Menezes AJ, Oorschot PC, Vanstone SA. *Handbook of Applied Cryptograph*. CRC Press: New York, 1997.
10. Wu L, Zhang Y, Wang F. A new provably secure authentication and key agreement protocol for SIP using ECC. *Computer Standards and Interfaces* 2009; **31**(2):286–291.
11. Yoon E, Shin Y, Jeon I, Yoo K. Robust mutual authentication with a key agreement scheme for the session initiation protocol. *IETE Technical Review* 2010; **27**(3):203–213.
12. Pu Q. Weaknesses of SIP authentication scheme for converged VoIP networks, 2010.
13. Tsai J. Efficient nonce-based authentication scheme for session initiation protocol. *International Journal of Network Security* 2009; **8**(3):312–316.
14. Arshad R, Ikram N. Elliptic curve cryptography based mutual authentication scheme for session initiation protocol. *Multimedia Tools and Applications* 2011; 1–14. DOI: 10.1007/s11042-011-0787-0.
15. Tang H, Liu X. Cryptanalysis of Arshad et al.'s ECC-based mutual authentication scheme for session initiation protocol. *Multimedia Tools and Applications* 2012; 1–13. DOI: 10.1007/s11042-012-1001-8.
16. Diffie W, Hellman ME. New directions in cryptography. *IEEE Transactions on Information Theory* 1976; **IT-22**:644–654.
17. Geneiatakis D, Dagiuklas T, Kambourakis G, Lambrinoudakis C, Gritzalis S, Ehlert S. Survey of security vulnerabilities in session initiation

protocol. *IEEE Commununication Survey Tutorials* 2006; **8**(3):68–81.
18. Veltri L, Salsano S, Papalilo D. SIP security issues: the SIP authentication procedure and its processing load. *IEEE Network* 2002; **16**(6):38–44.
19. Lee CC. On security of an efficient nonce based authentication scheme for SIP. *International Journal of Network Security* 2009; **3**:201–203.
20. Lu R, Cao Z. Off-line password guessing attack on an efficient key agreement protocol for secure authentication. *International Journal of Network Security* 2006; **3**(1):35–38.
21. Wang B, Li ZQ. A forward-secure user authentication scheme with smart cards. *International Journal of Network Security* 2006; **3**(2):116–119.
22. Bellare M, Pointcheval D, Rogaway P. Authenticated key exchange secure against dictionary attacks, Crypto00, 2000.
23. Boyko V, MacKenzie PD, Patel S. Provably secure password authenticated key exchange using Diffie–Hellman, Crypto00, 2000.
24. Abdalla M, Pointcheval D. Simple password based encrypted key exchange protocols, CT-RSA05.
25. Xie Q. A new authenticated key agreement for session initiation protocol. *International Journal of Communication Systems* 2011; **25**(1), doi:10.1002/dac.1286.
26. Callegari C, Garroppo RG, Giordano S, Pagano M. Security and delay issues in SIP systems. *International Journal of Communication Systems* 2009; **22**:1023–1044.
27. Debiao H, Jianhua C, Yitao C. A secure mutual authentication scheme for session initiation protocol using elliptic curve cryptography. *Security Communication Networks*, 2012; **5**(12):1423–1429.
28. Schneider B. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. New York: John Wiley & Sons, Incorporation, 1996.
29. Atkinson R. Security architecture for internet protocol, RFC 1825.
30. Rescorla E. *SSL and TLS: Designing and Building Secure Systems*. Addison-Wesley: NY, 2000.
31. Kent S, Atkinson R. Security architecture for the internet protocol, RFC 2401, Nov. 1998.
32. Ylonen T, Lonvick C. The secure shell (SSH) transport layer protocol, RFC 4253, Jan. 2006.
33. Raeburn K. Encryption and checksum specifications for Kerberos 5, RFC 3961, Feb. 2005.
34. Kilian J. A note on efficient zero-knowledge proofs and arguments. Proceedings for 24th Ann. ACM Symposium on Theory of Computing, Victoria, Canada, 1992; 723–732.
35. Shi Z, Beard C, Mitchell K. Analytical models for understanding space, Backoff and Flow Correlation in CSMA Wireless Networks, WIRELESS NETWORKS, Springer, DOI: 10.1007/s11276-012-0474-8, 2012.
36. Shi Z, Beard C, Mitchell K. Analytical Models for Understanding Misbehavior and MAC Friendliness in CSMA Networks. *Performance Evaluation*, September 2009; **66**(9–10): 469–487.
37. Ashraf CS, Nizamuddin, Sher M. Public verifiable signcryption schemes with forward secrecy based on hyperelliptic curve cryptosystem. *Communications in Computer and Information Science* 2012; **285**: 135–142,, DOI: 10.1007/978-3-642-29166-1_12.
38. Habib MA, Nasar W, Ashraf CS, Khan AJ. Ensuring minimal communication overhead in Low bandwidth network file, computer science and its applications. *Lecture Notes in Electrical Engineering* 2012; **203**:943–951.
39. Irshad A, Iqbal M, Ali A, Shafiq M. An algorithm for prediction of overhead messages in client–server based wireless networks. In *Computational Science and Its Applications-ICCSA*. Springer: Berlin Heidelberg, 2011; 412–423.