Multimed Tools Appl (2015) 74:3967–3984
DOI 10.1007/s11042-013-1807-z

A single round-trip SIP authentication scheme for Voice over Internet Protocol using smart card

Azeem Irshad · Muhammad Sher · Eid Rehman · Shehzad Ashraf Ch · Mahmood Ul Hassan · Anwar Ghani

Published online: 28 December 2013

Abstract The Session Initiation Protocol (SIP) has revolutionized the way of controlling Voice over Internet Protocol (VoIP) based communication sessions over an open channel. The SIP protocol is insecure for being an open text-based protocol inherently. Different solutions have been presented in the last decade to secure the protocol. Recently, Zhang et al. authentication protocol has been proposed with a sound feature that authenticates the users without any password-verifier database using smart card. However, the scheme has a few limitations and can be made more secure and optimized regarding cost of exchanged messages, with a few modifications. Our proposed key-agreement protocol makes a use of two server secrets for robustness and is also capable of authenticating the involved parties in a single round-trip of exchanged messages. The server can now authenticate the user on the request message received, rather than the response received upon sending the challenge message, saving another round-trip of exchanged messages and hence escapes a possible denial of service attack.

Keywords Session initiation protocol · Authentication · Security · Voice over Internet Protocol · Smart card

A. Irshad (✉) · M. Sher · Eid Rehman · S. A. Ch · M. U. Hassan · A. Ghani
Department of Computer Science & Software Engineering, Faculty of Basic and Applied Sciences,
International Islamic University, Islamabad, Pakistan
e-mail: irshadazeem2@gmail.com

M. Sher
e-mail: m.sher@iiu.edu.pk

Eid Rehman
e-mail: eidrehmanktk@gmail.com

S. A. Ch
e-mail: shahzad@iiu.edu.pk

M. U. Hassan
e-mail: mahmood-ul-hassan@iiu.edu.pk

A. Ghani
e-mail: anwar.ghani@iiu.edu.pk

1 Introduction

The voice over internet protocol (VOIP) based multimedia services are gaining a quick momentum, inducing a growth of internet telephony over traditional circuit-switched based telephony. There is an ease of deployment, maintenance, scalability, operation and above all, the economy, in the use of VOIP services [15]. The VOIP services require the use of a session initiation protocol (SIP) to create, maintain and terminate sessions [15, 23, 30]. This is a text based protocol and is derived from HTTP digest authentication [23], which is already susceptible to attacks. The SIP protocol makes a use of an insecure channel to deliver internet protocol (IP) packets to intended recipient. This calls for a robust security mechanisms, authentication and confidentiality in particular, to be followed for smooth functioning. Unfortunately, some of the areas received more of a focus than other ones, like quality of service over security.

The authentication can be performed in many ways for various applications like password-based authentication as one-time password [6], public-key cryptography, zero-knowledge proofs [20], digital signatures, and other authentication protocols such as Secure Socket Layer (SSL) [29], IP Security (IP SEC) [19], Secure Shell (SSH) [39] and Kerberos [28]. These authentication mechanisms depend upon various applications and the computing power. The SIP mutual authentication is based on the combination of password-based authentication and public key cryptography. Different authentication solutions have been presented in the last decade. Authenticated key agreement [1, 3–5, 31] requires the authenticity of corresponding participants to be ensured before initiating a call. The earlier solutions require the server to store all the users' passwords in its database. Then, the server has to protect this database of adversaries along with other things. This was obviously an additional burden on the server's useful resources. The problem was well taken by Zhang et al. [41] who came up with a smart card based solution. The solution obviates the need for the server to store the user passwords in some form. The user, during registration phase, sends its function modified password to server that stores it in smart card after applying another function to the message. Notwithstanding this intuition, the Zhang et al. technique was found vulnerable to a few threats as identified under, if properly fixed, the scheme could be further optimized to less costly and more secure solution.

In Zhang et al. technique [41] the user initiates the authentication mechanism by sending the request. The server in return would send the challenge after receiving the request. It keeps the calculated parameters intact until verification is finalized and the session is successfully established. The adversaries may exploit this scenario to present the simultaneous requests towards server and could deplete its resources if the session is not established. This is also known as denial of service (DOS) attack [18]. This aggravates the situation on the part of server, in peak hours and would not be able to meet its genuine user's needs, if such attacks are deployed. Secondly, the client's computing power varies from user to user, so does the key agreement time. If there are more than one round-trip of messages exchanged, there are chances of delay in session key agreement. Thirdly, a lot of solutions have been presented by taking the assumption of a single secret being compromised. Since, the robustness of a scheme does count if it still remains secure in spite of the compromise of a secret. These schemes didn't assume the compromise of all parameters. Alternatively, there are higher chances that a single secret may be exposed to some attacker rather than two or more. In Zhang et al. technique, the adversary after compromising the server secret may fully impersonate the server.

To counter the above threats, our proposed scheme completes the session creation in a single round trip, saving the resources and the probable DOS attack. The proposed scheme employs two secret keys on the side of server for the use in registration and authentication purposes. In our scheme, the server can authenticate the user on the first message received, rather than the response received on the second round-trip message.

The rest of the paper is organized as follows. In Section 2 the preliminaries defines the basics of ECC. The Section 3 presents the state-of-the art review of corresponding related techniques. The Section 4 shows the analysis of Zhang et al. scheme along with protocol working and drawbacks. The Section 5 describes the proposed model and Section 6 presents the security analysis of proposed model. The Section 7 illustrates the performance cost, while the last section concludes the findings.

2 Preliminaries

This section accommodates some of the basic elliptic curve cryptography (ECC) concepts pertinent to this paper. The ECC [7, 21, 26] security has been proved to be more efficient cryptographic scheme as compared to earlier conventional techniques [25] like RSA, DH and DSA. This technique provides an equivalent level of security with much less key sizes. The mathematical operations are defined over an elliptic curve equation

$$\mathrm{E_p(a,b) : y^2 = x^3 + ax + b(modp)\,and\,4a^3 + 27b^3 \neq 0(modp)},$$

Where $a, b \in F_p$ and 'p' be a large prime number. Both values a, b defines the elliptic curve, while the points (x, y) that satisfies the former statement including a point at infinity lies on the elliptic curve. The scalar multiplication is performed using $vP = P + P + ...P_v$, given a point P and an integer $v \in F_p^*$. All domain parameters like (p, a, b, G, n and h) belong to finite field, F_p^*. E is an abelian group and the point at infinity serves as identity element for this group. Here, we describe some of the security terms needed required to fully grasp the paper.

Term1: A Computational Diffie–Hellman Problem (CDHP) is stated as: Given three points P, aP, bP where $a,b \in F_{p,}^*$ it is hard to compute abP.

Term2: The Elliptic Curve Discrete Logarithm Problem (ECDLP) is stated as: given a point $Q = aP$ on Elliptic Curve, it would be hard enough to compute $a \in F_p^*$, given two points Q and P over $E(a,b)$.

Term3: The Elliptic Curve Factorization Problem (ECFP) is stated as: it is hard to find either aP or bP, given two points P and $Q = aP + bP$ over $E(a,b)$, while $a,b \in F_p^*$

Term4: A one-way hash operation as $y = h(x)$, where it is a hard problem to compute x, given y, in the above equation.

3 State-of-the-art review

Numerous authentication schemes have been proposed to date [2, 6, 8–11, 13, 14, 16, 17, 22, 24, 27, 30, 32–38, 40, 42] with various limitations [12]. In this regard, the first known authentication scheme, hyper text transfer protocol (HTTP) digest authentication based on RFC2617 [11], was unable to implement proper security mechanisms. Thereafter, a scheme [38] in 2005 proposed a SIP authentication technique but found victim to offline password-guessing and server spoofing attacks. This scheme utilized Diffie–Hellman key exchange algorithm based on the difficulty of Discrete Logarithm Problem (DLP). Another SIP authentication scheme [10] based on ECC [7, 21, 25, 26] was suggested in 2005. However, the scheme [10] suffered Denning-Sacco and stolen-verifier attacks. In 2009, the scheme [36] provides an enhanced level of security using ECC, and used Canetti–Krawczyk (CK) security model. Another scheme [40] in 2010 identified an offline password guessing attack in [36] scheme. The [40] scheme was presented for converged VoIP networks. Thereafter, the scheme [27] discovered a password guessing attack in [40]. Then, a study [33] in 2009, presented a technique based on hash and exclusive-OR (XOR) functions. The scheme [2] discovered a known-key secrecy, perfect forward secrecy, stolen-verifier and password-guessing attacks in [33] and proposed an enhanced protocol to defy the previous attacks. The scheme [8] proposed an authentication scheme that was also found under the same limitations that previous schemes suffered. The scheme [13] determined an offline-password guessing attack in [2] and presented an efficient protocol for SIP authentication.

Since, all of these schemes are based on storing users' passwords at server database and hence, are exposed to different attacks including stolen-verifier attack. Recently, a smart card based protocol by Zhang et al. [41] has been proposed to counter the identified threats associated with the earlier schemes. However, that scheme has the potential of further useful enhancements that contribute towards security and useful cost optimizations. We have tried to enhance security and optimize the cost in our proposed protocol by introducing some useful modifications.

4 Zhang et al. scheme analysis

The server secret plays a crucial role in the registration and authentication procedure, we will see lately how Zhang et al. scheme makes the use of the single secret its protocol. The Zhang scheme working and drawbacks have been described below.

4.1 Protocol working

1. In the Zhang et al. scheme, as shown in Fig. 1. the first message *REQUEST(username, X, Y)* is sent towards server after computing $X = bH + h(username)P$, $Y = bh(h(PW||a)||username)K_p$.
2. After receiving the Request message, the server computes $U = h(username)P$ and $Y' = s^2 (X - U)$, and verifies whether the equation holds $Y =_? Y'$. If so, then it selects two random integers $r \in_R Z_p^*$, $c \in_R Z_p^*$ and computes $R = cP$, $K = cs(X - U)$, $SK = h_1(K||r||username)$ and $Auth_s = h_2(K||Y'||r||SK)$, and sends a *CHALLENGE(realm, Auth$_s$, R, r)* message towards user. A realm is used to indicate the other participant about authentication protocol to be used.
3. The user now computes $K = bh(h(PW||a)||username)R$, $SK = h_1(K||r||username)$ and verifies whether the equation holds, i.e., $Auth_s =_? h_2(K||h(h(PW||a)||username)bK_p||r||SK)$. If so, then computes $Auth_u = h_2(K||h(h(PW||a)||username)bK_p||r + 1||SK)$ generates *RESPONSE(realm, Auth$_u$)* message and sends towards server.
4. The server now determines whether the equation holds $Auth_u =_? h_2(K\,||\,Y'\,||\,r + 1||SK)$. If true, then both entities treat *SK* as their mutually agreed session key and the protocol stops.

4.2 Drawbacks

The drawbacks of Zhang et al. scheme are as under:

1. Initially, when the user sends a *REQUEST(username, X, Y)* message towards server, an adversary may intercept the message and replay the message to server some other time. If so, the server will compute $U = h(username)P$ and $Y' = s^2(X - U)$, and verifies the equation $Y = ?\ Y'$. Since, the message does not contain any timestamp or freshness; the server will be forced to generate the challenge message. The server will only come to know about the validity of an adversary in the next step of response message generated by the adversary. A DOS attack might be launched against the server by an adversary in this manner. This attack can be thwarted if a proper timestamp or nonce message ensuring the message

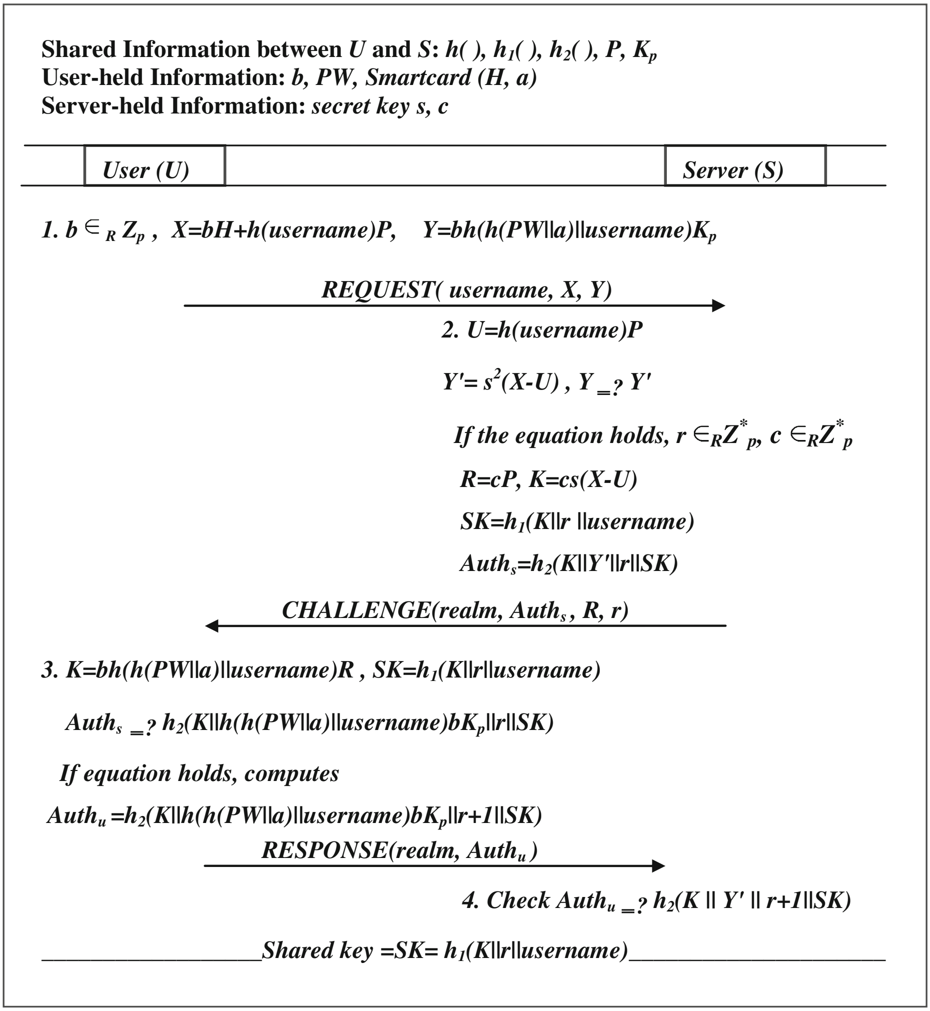

Shared Information between *U* and *S*: $h(\ ), h_1(\ ), h_2(\ ), P, K_p$
User-held Information: *b, PW, Smartcard (H, a)*
Server-held Information: *secret key s, c*
User (U)
Server (S)
1. $b \in_R Z_p$, $X=bH+h(username)P$, $Y=bh(h(PW||a)||username)K_p$
REQUEST(username, X, Y)
2. $U=h(username)P$
$Y'= s^2(X-U)$, $Y =_? Y'$
If the equation holds, $r \in_R Z^*_p$, $c \in_R Z^*_p$
$R=cP$, $K=cs(X-U)$
$SK=h_1(K||r\ ||username)$
$Auth_s=h_2(K||Y'||r||SK)$
CHALLENGE(realm, $Auth_s$, R, r)
3. $K=bh(h(PW||a)||username)R$, $SK=h_1(K||r||username)$
$Auth_s =_? h_2(K||h(h(PW||a)||username)bK_p||r||SK)$
If equation holds, computes
$Auth_u =h_2(K||h(h(PW||a)||username)bK_p||r+1||SK)$
RESPONSE(realm, $Auth_u$)
4. Check $Auth_u =_? h_2(K\ ||\ Y'\ ||\ r+1||SK)$
Shared key $=SK= h_1(K||r||username)$

Fig. 1 Zhang et al. authentication protocol

freshness is embedded in each Request message generated by the user, so that the message could not be replayed. Notwithstanding the fact, that the server would come to know eventually about this attack when it receives the response from the adversary, yet the attacker could affect the efficient working of server by launching a DOS attack that may deplete the server useful resources.

2. In a smart card based system the role of a secret key has been even more important than earlier schemes, since, in earlier schemes if secret key of server has been compromised, an attacker would also require the password-verifier database as well to access the users' passwords. But in Zhang scheme, the access of server secret to an adversary directly approaches the users' passwords. In this scenario, the use of a single secret might expose the whole system if compromised.
3. The Zhang authentication protocol could be reduced to a single-round trip protocol rather than a one and a half round-trip as used by it.

5 Proposed scheme

The theft of a single secret could be possible accidentally that may jeopardize the whole system; however, the system can be made more robust if two server secrets are employed to authenticate the users with server. The proposed scheme presents an authenticated key agreement protocol between user and server to optimize and counter the flaws in Zhang et al. scheme. The proposed scheme focuses on the completion of authentication phase in a single round-trip phase and to improve the efficiency of protocol. The authentication is performed between two entities, smart card (user) and the server. The protocol consists of different phases for key agreement: system setup phase, registration phase, authentication phase, and password updating phase.

5.1 System setup phase

In this system setup phase, different parameters are defined that will be considered to be available for public use or user's interaction with the system. A few steps are described below undertaken by the server for setting up the system.

Step 1. The server selects an elliptic curve equation $E_P(a,b)$ with the order n.

Step 2. A base point P of order n is selected by server over an elliptic curve equation Ep(a,b), where n is a large number of high entropy. The server picks two secret keys $s_1 \in_R Z_p$ and $s_2 \in_R Z_p$.

Step 3. The server chooses three one-way hash functions $h(), h_1()$ and $h_2()$. It also selects its public key as $K_p=s_2P$ using the second secret key s_2 and then publishes all of the above information.

5.2 Registration phase

In registration phase, the server verifies the user through a secure channel.

Step 1. On positive verification, the user chooses a password PW and a random integer $a\in_R Z_p^*$. Afterwards, it computes $h(PW||a)$ and sends $h(PW||a)||username$ and a randomly generated key E to server using a secure channel.

Step 2. After that, the server computes $H=h(h(PW\|a)\|username)s_2^{-1}P$ and $\mathrm{I}=(E\|username)s$. Now, the server stores H in smart card and sends this card and I to user using a secure channel.

Step 3. The user stores the nonce a in the smart card. The memory of smart card now contains (H, a).

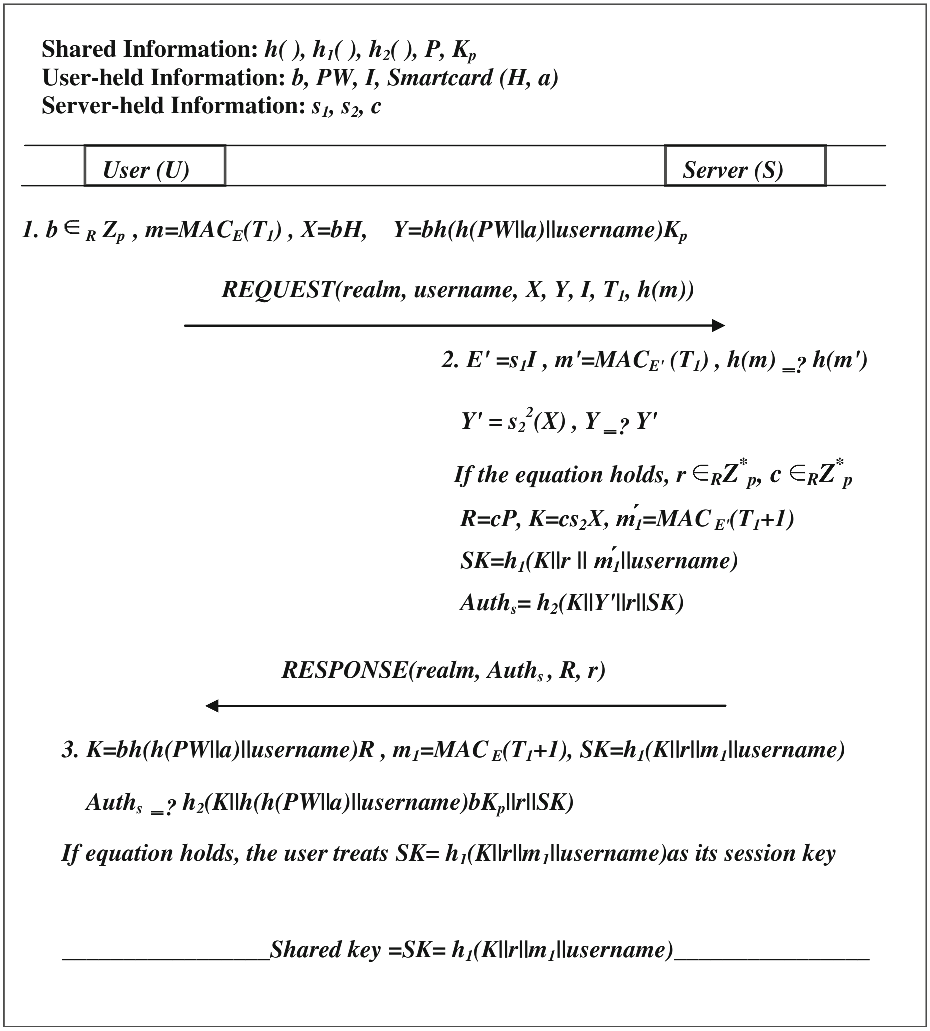
5.3 Authentication phase

Whenever the user U tries to log into the server, it uses its card through smart card reader. The user also inputs its username and password PW. Afterwards, the user and server are authenticated using the following protocol as shown in Fig. 2.

Step 1: Initially, the user chooses a random integer $b\in_R Z_p$ and a timestamp T_1. Next, it computes $X=bH$, $Y=bh(h(PW||a)||username)K_{p.}$ and $m=MAC_E(T1)$. Afterwards, it sends *REQUEST(realm, username, X, Y, I, T_1, h(m))* to server using the public channel.

Step 2: The server, after receiving the request, computes $E'=s_1I$ *and* $m'=MAC_{E'}\ (T_1)$ and verifies that whether *h(m)* is equal to *h(m')*. The successful verification authenticates the freshness of timestamp and message. Next, the server computes $Y'=s_2^2(X)$. and verifies whether $Y \underset{?}{=} Y'$. After successful verification, it chooses two random integers c and r to compute $R=cP$, $K=csX$, $_1=MAC_{E'}(T_1+1)$, $SK=h_1(K||r||\acute{m}_1||username)$ and finally $Auth_s=h_2(K||Y'||r||SK)$. The server, after doing verification, sends a response *RESPONSE(realm, $Auth_s$, R, r)* to prove its identity towards user.

Step 3: The user receives the response and computes $K=bh(h(PW||a)||username)R$, $m_1=MAC_E(T_1+1)$, and $SK=h_1(K||r||m_1||username)$. Then, it confirms that whether the calculated parameter $h_2(K||h(h(PW||a)||username)bK_s||r||SK)$ equates the received



Shared Information: $h(\,), h_1(\,), h_2(\,), P, K_p$
User-held Information: b, PW, I, *Smartcard (H, a)*
Server-held Information: s_1, s_2, c
User (U)
Server (S)
1. $b \in_R Z_p$, $m=MAC_E(T_1)$, $X=bH$, $Y=bh(h(PW||a)||username)K_p$
REQUEST(realm, username, X, Y, I, T_1, h(m))
2. $E'=s_1I$, $m'=MAC_{E'}\ (T_1)$, $h(m) \underset{?}{=} h(m')$
$Y'=s_2^2(X)$, $Y \underset{?}{=} Y'$
If the equation holds, $r \in_R Z^*_p$, $c \in_R Z^*_p$
$R=cP$, $K=cs_2X$, $m'_1=MAC_{E'}(T_1+1)$
$SK=h_1(K||r\ ||\ m'_1||username)$
$Auth_s= h_2(K||Y'||r||SK)$
RESPONSE(realm, $Auth_s$, R, r)
3. $K=bh(h(PW||a)||username)R$, $m_1=MAC_E(T_1+1)$, $SK=h_1(K||r||m_1||username)$
$Auth_s \underset{?}{=} h_2(K||h(h(PW||a)||username)bK_p||r||SK)$
If equation holds, the user treats $SK= h_1(K||r||m_1||username)$as its session key
Shared key $=SK= h_1(K||r||m_1||username)$

Fig. 2 Proposed authentication model

$Auth_s$. If it validates the equality, then it proceeds with the shared session key SK, otherwise, it stops the protocol and deletes the calculated parameters.

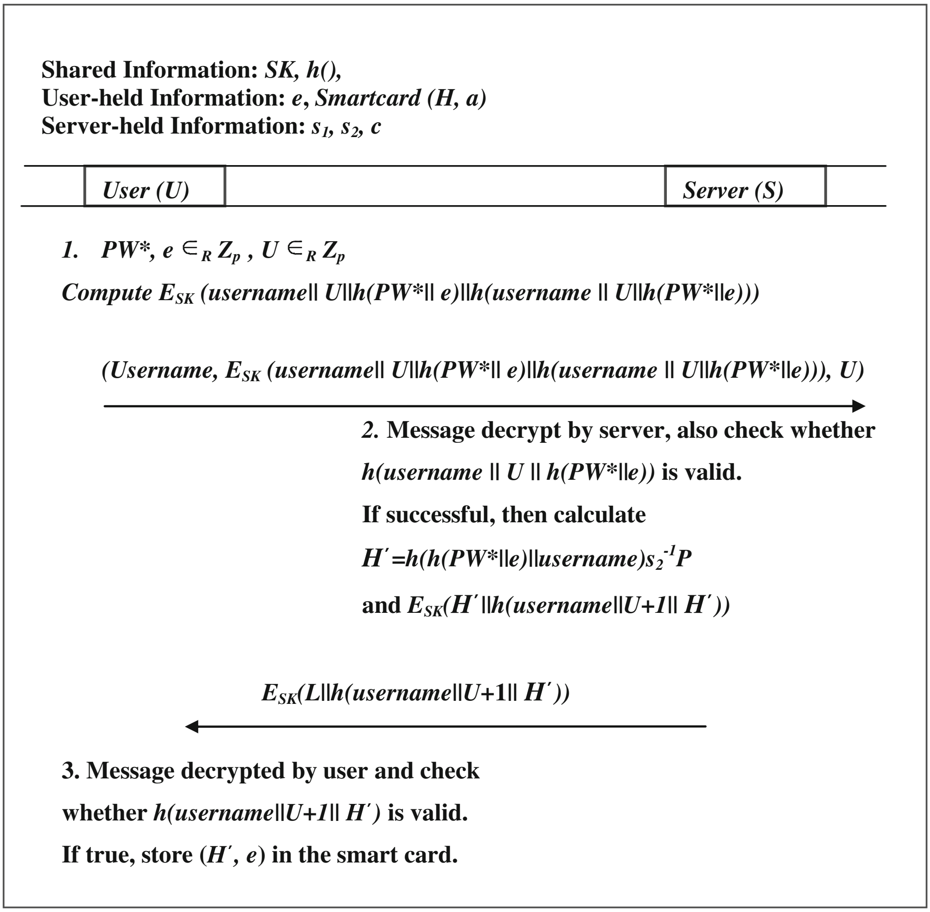
5.4 Password updating phase

A current session key *SK* is used to initiate the password updating procedure [41]. This procedure has been shown in Fig. 3 and explained below.

Step 1. The user selects a new password *PW** and a random integer $e \in_R Z_p$. The session key *SK* is used to encrypt the message that consists of the new password. The encrypted message E_{SK} *(username|| U||h(PW*|| e)||h(username || U||h(PW*||e)))* is sent to the server along with the timestamp *U* or checking the message freshness.

$$U \rightarrow S : (Username, E_{SK}(username \| U \| h(PW * \| e) \| h(username \| U \| h(PW^* \| e))), U)$$

Step 2. The server decrypts the message after receiving, and verifies its authenticity using *h(username || U || h(PW*||e))*. If successful, the server would calculate the new



Shared Information: ***SK, h(),***
User-held Information: ***e, Smartcard (H, a)***
Server-held Information: s_1, s_2, c

User (U) ***Server (S)***

1. $PW^*, e \in_R Z_p$, $U \in_R Z_p$

Compute E_{SK} ***(username|| U||h(PW*|| e)||h(username || U||h(PW*||e)))***

(Username, E_{SK} ***(username|| U||h(PW*|| e)||h(username || U||h(PW*||e))), U)***

2. **Message decrypt by server, also check whether *h(username || U || h(PW*||e))* is valid.**

If successful, then calculate

$H' = h(h(PW^* \| e) \| username) s_2^{-1} P$

and $E_{SK}(H' \| h(username \| U+1 \| H'))$

$E_{SK}(L \| h(username \| U+1 \| H'))$

3. Message decrypted by user and check whether *h(username||U+1|| H′)* is valid.

If true, store (*H′*, *e*) in the smart card.

Fig. 3 Password updating phase

parameter as $H' = h(h(PW^{*}||e)||username)s_2^{-1}P$ and sends the message after encryption $E_{SK}(H'||h(username\ ||U+1||\ H'))$ towards user.

$$S \rightarrow U : E_{SK}(H'||h(username||U+1||H'))$$

Step 3. The user decrypts the received message and verifies its authenticity by checking $h(username||U+1||\ H')$ and stores $(H',\ e)$ in the smart card on positive verification.

6 Security analysis

The security analysis of the proposed protocol has been presented as under:

6.1 Replay attacks

The replay attacks are launched when an adversary A replays the genuine message parameters at some other time to deceive or impersonate any legitimate participant. The proposed scheme is resistant to such attacks, if A tries to reuse the contents of $REQUEST(realm,\ username,\ X, Y, I, T_1,\ h(m))$ message, since the scheme makes a use of timestamps. The server instant comes to know about the genuineness of timestamp after taking The adversary cannot generate m, as the $m = MAC_E(T_1)$ can only be retrieved by the entity having E. The E can only be recovered by the server using its first secret key s_1. The parameter $X = bH$ and $Y = bh(h(PW||a)||username)K_p$ cannot be reconstructed since it requires knowledge of s_2 and will face ECDLP to recover. The server can thwart the attack by confirming the equivalence of Y with Y' while $Y' = s_2^2(X)$.

The adversary may intercept the $RESPONSE(realm,\ Auth_s,\ R,\ r)$ and try to impersonate server. The user computes $h_2(K||h(h(PW||a)||username)bK_p||r||SK)$ by calculating three parameters as $K = bh(h(PW||a)||username)R$, $m_1 = MAC_E(T_1 + 1)$, and $SK = h_1(K||r||m_1||username)$ to compare with $Auth_s$. The user authenticates the server on positive verification and rejects the message otherwise. The A needs H and b values to determine bH and compute K, which are inaccessible to it.

6.2 Man in the middle attack

This attack is launched by A to act as 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 A if the attack is successful.

For this attack to be effective, the adversary needs to generate the same session key SK as is shared by the intended participants. However, in the proposed protocol, the A needs to access either bH or the parameters like c or s_2 . However it faces ECDLP to recover c from R. Hence, we can say, the proposed protocol can rightly defend the Man in the Middle Attack launched either on user or server.

6.3 Modification attacks

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

The A might generate a modified *REQUEST(realm, username,* $X', Y', I', T_1', h'(m')$ *)* message. To counter the threat, the server applies the multiplication operation of $\acute{I}$ with s_1 and gets E'' i.e., $E'' = s_1 \acute{I}$ which is further used to derive $m'' = MAC_{E''}(\acute{T}_1)$. Next it examines that whether $h(m'')$ matches with the received $h(\acute{m})$. On successful verification it would proceed with a confidence to escape a possible modification attack.

If A tries to generate a *RESPONSE(realm,* $Auth'_s, \acute{R}, \acute{r}$ *)* message and sends towards user with an intent to impersonate server, the user may thwart the attack by calculating the parameters $\acute{K} = bh(h(PW\|a)\|username)\acute{R}$, $m_1 = MAC_E(T_1 + 1)$ and $S\acute{K} = h_1\left(\acute{K}\|\acute{r}\|m_1\|username\right)$ and verifying $Auth'_s$ with its calculated $h_2\left(K\Big\|h(h(PW\|a)\|username)bK_p\|r\|S\acute{K}\right)$. The verification process would fail if an attacker A had generated the $Auth'_s$ parameter either without knowledge of the s_2 key or the valid *PW, a* and *b* values.

6.4 Denning-Sacco attack

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

In proposed model, the session key is generated by taking hash i.e., $SK = h_1(K\|r\|m_1\|username) = h_1(bh(h(PW\|a)\|username)cP\ \|r\|m_1\|\ username)$. If an attacker is able to compromise an old session key, however, it cannot derive *PW, a, b* and *c* out of the old session key and will have to face the ECDLP and break the hash function as well. Hence, the proposed scheme can resist the Denning-Sacco attacks.

6.5 Stolen verifier attacks

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

In the proposed model, there is no such user's information maintained at server that can be stolen to the attackers benefit. Hence, the proposed protocol can rightly defend against the stolen verifier attack since there is no verifier stored for verification of users.

6.6 Offline dictionary threat without using smart card

In an offline dictionary threat an attacker tries to guess the secret parameters out of intercepted messages without using the smart card. In the proposed protocol if an attacker intercepts the contents of *REQUEST(realm, username, X, Y, I,* T_1*, h(m))* message, first in $X = bH$ parameter, it has to recover *E* for getting *bH* and then face ECDLP to get *H*. Likewise, in parameter $Y = bh(h(PW\|a)\|username)K_p$ the attacker needs to face ECDLP for recovering either *b* or $h(h(PW\|a)\|username)$ in *Y.* The $Auth_s$ tag in *RESPONSE(realm,*$Auth_s$*,R,r)* comprise three parameters *K, Y' and SK* which need to be guessed for the offline attack to activate. Hence, the proposed scheme is invulnerable to offline dictionary threats without using smart cards.

6.7 Offline dictionary threat using smart card

In offline dictionary threat, an attacker steals a smart card and tries to use the derived information with the input of all possible combinations of guessed secrets by applying brute force attack.

In proposed protocol, the smart card bears two parameters *(H, a)* where $H = h(h(PW \| r) \| username) s_2^{-1} P$. An attacker needs to extract $h(h(PW||r)||username)$ either from H or $Auth_s$ to launch an offline dictionary attack, which will have to face elliptic curve discrete logarithm problem. Hence, the proposed scheme is immune to possible threats likely to trigger in the wake of illegitimate smart card acquisition.

6.8 Session key security

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

In proposed protocol, the session key $SK = h_1(K||r||m_1||username) = h_1(bch(h(PW||a)|| username)P\ ||r||\ MAC\ _E(T_1 + 1)||username)$ comprise the hash of combinations of *a, b, c, PW* and key hashed $T_1 + 1$ parameters that needs to be determined by an attacker for generating an exact session key. The *SK* can only be generated by a legitimate user and a server, using the proposed protocol.

6.9 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.

In proposed protocol, the session key $SK = h_1(K||r||m_1||username) = h_1(bch(h(PW||a)|| username)P\ ||r||\ MAC\ _E(T_1 + 1)||username)$ is generated out of *a, b, c* parameters. The first two are used by the user while the last one by server for a unique session key generation. These parameters are used by the participants independently in the exchanged messages using hash digest, which helps hiding it of anyone even from the other participant. Each run of the authentication protocol generates a unique session key, since, the session parameters like *b* and *c* are randomly selected and different each time. If an attacker comes to know about the *SK*, *b* and *c* for any session, it cannot guess either *SK* or *b* or *c* for another session. Hence, the proposed scheme provides the known-key security to the communicating participants.

6.10 Perfect forward secrecy

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

In the proposed protocol, if the user's password *PW* or server's secret keys i.e., either s_1 or s_2 or both server keys are compromised then an attacker cannot recover the previous session keys by using the intercepted messages as it will have to face ECDLP to recover c from $R = cP$, *a* and *b* from $X = bH$ and $Y = bh(h(PW||a)||username)K_p$. Hence, the proposed scheme provides the perfect forward secrecy.

6.11 Mutual authentication

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

In the proposed protocol the server authenticates the user first by dual authentication i.e., initially by verifying timestamp $h(m) =_? h(m')$ after evaluating the parameters $E' = s_1 I$, $m' = MAC_{E'}(T_1)$. Later, the server evaluates the parameters and again verifies $Y =_? Y'$, that ensures the user's authentication. Secondly, the user authenticates the server by verifying the received parameter *Auths* with a computed value $h_2(K||h(h(PW||a)||username)bK_p||r||SK)$, respectively. Hence, the proposed protocol provides mutual authentication.

6.12 Secure password update

The proposed scheme binds the user to use smart card if it wants to update its password. A user may update its password without any restraint in the registration phase using the current session key *SK* if it possess the legal smart card. The smart card obviates the need for remembering password before use. However, if a user forgets its password, it might update its password using the smart card and the current session key.

7 Comparison and cost analysis

In this section the security and efficiency analysis has been presented that compares proposed authentication model with the Zhang et al. protocol. Here are a few notations used in this section, each representing the computation cost of a single operation in terms of time.

T_{ESM} being the time for performing elliptic curve scalar multiplication
T_{EPA} being the time for performing elliptic curve point addition operation.
T_H being the time for executing a one-way hash operation.
T_{INV} being the time for performing a modular inversion operation.
T_{KH} being the time for performing keyed hash operation, also known as MAC.
T_{ES} being the time for executing symmetric key encryption operation.
T_{DS} being the time for taking symmetric key decryption operation.

The security has been directly related to cost optimization, since the increase in security boost up the cost and vice-versa. However, in the proposed scheme, the security has been enhanced with a reduced cost. Whenever comparing different security protocols, we put more focus on few operations for comparison since these operations are more costly than others. For example, T_{ESM}, being the scalar multiplication operation, takes more computation cycles than other ones; therefore, the tendency must be to reduce the number of T_{ESM} operations in the construction of protocol to a level such that the security is not compromised. Hence, T_{ESM} is now considered as more significant for comparing the overhead cost of different authentication protocols. At the same time, the time for hash (T_H) keyed hash (T_{KH}) and point addition operations (T_{EPA}) also matter in comparison, but as a secondary status, since the later operations taking fewer computation cycles.

Table 1 Comparison between Zhang et al., and proposed protocol

Schemes Types of messages	Zhang et. al. protocol	Proposed protocol
Registration messages	1 T_{ESM} +1 T_H+1 T_{INV}	1 T_{ESM} +1 T_H+2 T_{INV}
Authentication messages	9 T_{ESM} +2 T_{EPA} +10 T_H	7 T_{ESM} +8T_H+4 T_{KH}
Password update messages	2 T_{ES} +2 T_{DS}+1 T_{ESM}+6 T_H+1 T_{INV}	2 T_{ES} +2 T_{DS}+1 T_{ESM}+6 T_H+1 T_{INV}
Total messages	11 T_{ESM} +17 T_H+2 T_{INV}+2 T_{EPA} +2 T_{ES} +2 T_{DS}	9 T_{ESM}+19 $T_{H/KH}$+3 T_{INV}+2 T_{EPA} +2 T_{ES} +2 T_{DS}

Table 2 Attacks on protocols under different conditions

Schemes Threats/Roundtrips	Zhang et al. scheme	Proposed scheme
1. Modification attack	S	S
2. Man in the middle attack	S	S
3. Replay attack	S	S
4 Mutual Authentication	S	S
5. Denial of Service attacks	IS	S
6. Known key security	S	S
7. Perfect forward security	S	S
8. Session key secrecy	S	S
9. Stolen verifier attack	S	S
10. Denning sacco attack	S	S
11. Single round trip of protocol	NP	P

S secure, *IS* insecure, *NP* not provided, *P* provided

The Zhang et al., scheme comprises 9 T_{ESM} +2 T_{EPA} +10 T_H authentication messages, while the proposed scheme $7T_{ESM}$ $+8T_H$+$4T_{KH}$ messages. The Zhang et al., scheme incurs 9 scalar multiplication computations (T_{ESM}), 2 elliptic curve point additions (T_{EPA}) and 10 hash operations (T_H) in a single run of protocol. On the other hand, the proposed protocol incurs 7 T_{ESM}, 8 $\mathrm{T_H}$ and 4 keyed hash (T_{KH}) operations. Here we can see that the proposed protocol incurs two less T_{ESM} operations in comparison with Zhang scheme. The registration messages are almost the same except one additional modular inversion operation in proposed scheme as compared to Zhang scheme, which is negligible. The password changing procedure has been same for both of the schemes (Table 1).

Hence, in the light of above performance analysis, we can say that the proposed scheme is more efficient than Zhang scheme. The Table 2 summarizes the attacks on the Zhang and proposed protocol, which shows the Zhang scheme might suffer denial of service attack on both ends, because end parties need to maintain the variables during the session that could be exploited by an adversary. The proposed scheme defends the denial of service threat, provides enhanced security, cost effective and maintains reliability.

We have used Automated Validation of Internet Security Protocols and Applications (AVISPA) to study the effects of reduced computation cost and call delay in the related protocols. The proposed protocol leads to 40 % saving in average computation cost on end points and 33 % call delay reduction due to smarter round-trip time than before. The two less elliptic curve scalar multiplications in proposed protocol as compared with Zhang protocol, contribute to these computational savings. In the future work the performance will be formally evaluated and presented. Further, efficient ways to authenticate the relevant entities will be presented and analyzed.

8 Conclusion

In this paper, we have determined several weaknesses in Zhang et al. scheme that includes particularly, a possible denial of service attack, which is addressed in our proposed model. Firstly, our proposed protocol authenticates the entities in a single round-trip. Secondly, it improves the security by the implementation of two server secrets used simultaneously for user registration and authentication purposes, and escaping the possible denial of service attack. Hence, it makes an efficient use of server resources by not engaging itself in the establishment of partial sessions without authenticity, so that it needs not reserving resources for variables of partially established sessions. In Zhang scheme, an accidental leakage of secret would render the system in a fiasco. The use of two secrets in proposed scheme ensures the robustness in a way that an accidental leakage of either of the two secrets would not expose the whole system. Thirdly, our scheme employs fewer scalar multiplications than Zhang scheme which is the real indicator for cost comparison.

References

1. Abdalla M, Pointcheval D (2005) Simple password based encrypted key exchange protocols (CT-RSA 2005)
2. Arshad R, Ikram N (2011) Elliptic curve cryptography based mutual authentication scheme for session initiation protocol. Multimed Tools Appl. doi:10.1007/s11042-011-0787-0
3. Atkinson R. Security architecture for the Internet protocol, RFC 1825
4. Bellare M, Pointcheval D, Rogaway P (2000) Authenticated key exchange secure against dictionary attacks (Crypto 2000)
5. Boyko V, MacKenzie PD, Patel S (2000) Provably secure password authenticated key exchange using diffie-hellman (Crypto 2000)
6. Callegari C, Garroppo RG, Giordano S, Pagano M (2009) Security and delay issues in SIP systems. Int J Commun Syst 22:1023–1044
7. Certicom Research Standard for efficient cryptography, SEC 1, 2000: EC Cryptography. Ver. 1.0
8. Debiao H, Jianhua C, Yitao C (2012) A secure mutual authentication scheme for session initiation protocol using elliptic curve cryptography, Published online in Wiley Online Library wileyonlinelibrary.com. Security Comm Netw. doi:10.1002/sec.506
9. Diffie W, Hellman ME (1976) New directions in cryptography. IEEE Trans Inf Theory IT-22:644–654
10. Durlanik A, Sogukpinar I (2005) SIP authentication scheme using ECDH. World Enformatika Soc Trans Eng Comput Technol 8:350–353
11. Franks J, Hallam-Baker P, Hostetler J, Lawrence S, Leach P, Luotonen A, Stewart L (1999) HTTP authentication: basic and digest access authentication, IETF RFC2617
12. 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
13. Hongbin T, Xinsong L (2012) Cryptanalysis of Arshad et al.'s ECC-based mutual authentication scheme for session initiation protocol. Multimed Tools Appl. doi:10.1007/s11042-012-1001-8
14. Huang H, Wei W, Brown G (2006) A new efficient authentication scheme for session initiation protocol. Proceedings of JCIS 06
15. Hussain TH, Marimuthu PN, Habib SJ (2012) Supporting multimedia applications through network redesign. Int J Commun Syst. doi:10.1002/dac.2371
16. Irshad A, Noshairwan W, Shafiq M, Khurram S, Irshad E, Usman M (2008) Security enhancement in MANET authentication by checking the CRL status of servers. Int J Adv Sci Technol 1:91–98
17. Jo H, Lee Y, Kim M, Kim S, Won D (2009) Off-line password-guessing attack to Yang's and Huang's authentication schemes for session initiation ptorocol. Proceedings of INC, IMS and IDC, pp 618–621
18. Karig D, Lee R (2001) Remote denial of service attacks and countermeasures. Department of Electrical Engineering, Princeton University, Technical Report CE-L2001-002
19. Kent S, Atkinson R (1998) Security architecture for the Internet protocol, RFC 2401
20. Kilian J (1992) A note on efficient zero-knowledge proofs and arguments. In: Proc. 24th Annual ACM Symposium on Theory of Computing, Victoria BC, pp 723–732
21. Koblitz N (1987) Elliptic curve cryptosystems. Math Comput 48:203–209
22. Lee CC (2009) On security of an efficient nonce based authentication scheme for SIP. Int J Netw Secur 3:201–203
23. Li J-S, Kao C-K, Tzeng J-J (2011) VoIP secure session assistance and call monitoring via building security gateway. Int J Commun Syst 24:837–851
24. 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

25. Menezes AJ, Oorschot PC, Vanstone SA (1997) Handbook of applied cryptograph. CRC Press, New York
26. Miller V (1986) Uses of elliptic curves in cryptography. In: Advances in cryptology CRYPTO'85, Lecture Notes in Computer Science, vol. 218. Springer-Verlag, pp 417–426
27. Pu Q (2010) Weaknesses of SIP authentication scheme for converged VoIP networks, http://eprint.iacr.org/2010/464
28. Raeburn K (2005) Encryption and checksum specifications for Kerberos 5, RFC 3961
29. Rescorla E (2000) SSL and TLS: designing and building secure systems. Addison-Wesley, New York
30. Rosenberg J, Schulzrinne H, Camarillo G, Johnston A, Peterson J, Sparks R, Handley M, Schooler E (2002) SIP: session initiation protocol, IETF RFC3261
31. Schneider B (1996) Applied cryptography second edition: protocols, algorithms, and source code in C. John Wiley & Sons Inc., Hoboken
32. Thomas M (2001) SIP security requirements. IETF internet draft (draftthomas-sipsec-reg-00.txt), work in progress
33. Tsai J (2009) Efficient nonce-based authentication scheme for session initiation protocol. Int J Netw Secur 8(3):312–316
34. Veltri L, Salsano S, Papalilo D (2002) SIP security issues: the SIP authentication procedure and its processing load. IEEE Netw 16(6):38–44
35. Wang B, Li ZQ (2006) A forward-secure user authentication scheme with smart cards. Int J Netw Secur 3(2):116–119
36. Wu L, Zhang Y, Wang F (2009) A new provably secure authentication and key agreement protocol for SIP using ECC. Comput Stand Interfaces 31(2):286–291
37. Xie Q (2011) A new authenticated key agreement for session initiation protocol. Int J Commun Syst. doi:10.1002/dac.1286
38. Yang C, Wang R, Liu WT (2005) Secure authentication scheme for session initiation protocol. Comput Secur 4:381–386
39. Ylonen T, Lonvick C (2006) (Eds) The secure shell (SSH) Transport layer protocol, RFC 4253
40. Yoon EJ, Koo KY (2010) Robust mutual authentication with a key agreement scheme for the session initiation protocol. IETE Tech Rev 27(3):203–213
41. 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 Comm Syst
42. Zhou L, Chao H-C, Vasilakos A (Aug. 2011) Joint forensics-scheduling strategy for delay-sensitive multimedia applications over heterogeneous networks. IEEE J Sel Areas Commun 29(7):1358–1367

Azeem Irshad is a PhD scholar at International Islamic University Islamabad Pakistan. He has done MS-CS from Arid Agriculture University Rawalpindi. Besides, he is a lecturer at Govt. Postgraduate College, Attock, Pakistan. He has got more than 14 research publications in international journals and conferences. His research interests include SIP security, key agreement and authentication, Elliptic curve cryptography, security, MANETs, Next Generation Networks, LTE security and mobility.

Dr. Muhammad Sher is a Professor having more than 120 scientific publications. He is chairman of the Department of Computer Science & Software Engineering, International Islamic University. He is also Dean of the Faculty of Basic & Applied Sciences. He did his Ph.D. Computer Science from TU Berlin, Germany and M. Sc. from Quaid-e-Azam University, Islamabad. His research interests include Next Generation Networks and Network Security.

Eid Rehman is currently a PhD scholar at International Islamic University Islamabad Pakistan. He received his MS degree in computer science from the International Islamic University Islamabad, Pakistan, in October 2012. His research interests Next Generation Network, IP Multimedia Subsystem, wireless sensor networks, and mobile ad hoc network. Currently, he is working as a Research Professional in ICT R&D project in International Islamic University Islamabad, Pakistan.

Shehzad Ashraf Ch received distinction in his Master degree from International Islamic University Islamabad, Pakistan in 2009. He was also awarded Gold Medal for achieving 4.0/4.0 CGPA in his Masters, Currently working as Lecturer in IIUI, He is also a PhD candidate of the same institute. His research interests include Lightweight Cryptography, Elliptic/Hyper Elliptic Curve Cryptography, Multimedia Security, MANETs, SIP authentication, IP Multimedia sub-system and Next Generation Networks. He has published more than 20 research papers in International Journals and Conferences.

Mahmood Ul Hassan is currently a PhD scholar at International Islamic University Islamabad Pakistan. He has been working as Assistant Director at Department of Computer Science & Software Engineering, International Islamic University, Islamabad, Pakistan. His research interests Next Generation Networks, quality of service, IP Multimedia Subsystem, wireless sensor networks, and mobile ad hoc network.s.

Anwar Ghani received the Bachelor degree from University of Malakand K.P.K, Pakistan in 2007, MSCS degree from International Islamic University Islamabad, Pakistan in 2011. He is currently a Ph.D. student in the Department of Computer Science & Software Engineering, International Islamic University, Islamabad, Pakistan. His research interests include network security and wireless sensor networks.