Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2016) 24: 3158 – 3177

doi:10.3906/elk-1404-133

Analyzing the mutual authenticated session key in IP multimedia server-client systems for 4G networks

Bakkiam David DEEBAK[1,*], **Rajappa MUTHAIAH**[1], **Karuppuswamy THENMOZHI**[2], **Pitchai Iyer SWAMINATHAN**[1]

[1]School of Computing, SASTRA University, Thanjavur, Tamil Nadu, India

[2]School of Electrical and Electronics Engineering, SASTRA University, Thanjavur, Tamil Nadu, India

Received: 08.04.2014 • **Accepted/Published Online:** 13.03.2015 • **Final Version:** 15.04.2016

Abstract: This paper scrutinizes the authentication and key agreement protocol adopted by the Universal Mobile Telecommunication System to meet the standards of a fourth-generation network. Lately, communication of multimedia (CoM) has drawn the attention of researchers for the future of secure wireless mobile communication. However, the CoM has not had any defensive mechanism to fulfil the specifications of 3GPP and reduce the computation and communication overheads and susceptible attacks like redirection, man-in-the-middle, and denial of service attacks. In addition, this paper has thoroughly investigated some existing protocols from the literature for the identification of new challenges in server-client authentication. To probe the challenges of the existing schemes realistically, the multimedia client and multimedia server components (proxy, interrogating, serving, and home subscriber server) were physically deployed on the Linux platform to examine the specifications of 3GPP, vulnerable attacks, computation, and communication overheads. We observed that the examined existing schemes are not able to fulfill the above criteria. We thus propose addition of the mutual authenticated session key (MASK) to the physical environment of the multimedia server-client. To satisfy the 3GPP specifications, the protocol of MASK offers mutual authenticity to the multimedia server-client. Moreover, the feature of mutual authenticity reduces the computation and communication overheads of the multimedia server-client. Since the session keys are jointly shared between the multimedia server and client, the protocol of MASK can additionally provide privacy preservation and forward secrecy.

Key words: Universal Mobile Telecommunication System, communication of multimedia, secure wireless mobile communication, 3GPP, authentication and key agreement, mutual authenticated session key, multimedia server-client

1. Introduction

The Universal Mobile Telecommunication System (UMTS) standardized the fastest third-generation (3G)-based systems for the technology of mobile communication. For the fourth-generation (4G) network, it took up the authentication and key agreement (AKA) protocol, which was designed to ensure the secure provisional services of multimedia like voice, video, and instant messaging over the Internet [1–4]. However, it has not had any counteracting mechanisms for packet sniffers and flooding attackers. Thus, the packet contents of multimedia cannot be secured over the Internet. In the first-generation (1G) network, the challenging issues of security were not as remarked as they should have been and thus, with the use of low-cost technology, anomalies can overhear the users' traffic to exploit services. To resolve the challenging issues of 1G, the second-generation (2G) network implemented the Global System for Mobile Communication (GSM).

*Correspondence: jrvd_deebak@ymail.com

Unfortunately, the authentication of the GSM was unidirectional and thus failed to authenticate the serving networks. The lack of mutual authenticity of the 2G network has hence brought the issue of false base-station attacks. To make the authentication bidirectional, the protocol of 3GPP AKA has emerged as GSM AKA for significant goals such as mutual authentication, agreement on an integrity key, and assurance on the cipher and integrity keys. The purpose of AKA protocol is to use a key generation mechanism-based challenge-response to ensure whether security properties are satisfied or not.

The objective of the GSM AKA was to generate the authentication vector to achieve mutual authentication over the users and serving networks. Then the generated authentication vectors of the users and serving networks are checked for identity matching. If the matching is successful, then the users get the connection through the serving network to access the services of GSM. Otherwise, the users and serving networks need resynchronization to adjust the authentication vector in the home network. 3GPP collaborates with the telecommunication group to introduce a 3G mobile system. To date, the AKA security mechanism of 3GPP has had many serious flaws over public networks. To be flawless, the traditional cryptosystem has been adapted as a public key cryptosystem. Since mobile devices have limited power and computational capability, they do not support the public key cryptosystem.

To determine the solution, an elliptic-curve cryptography (E-CC) technique is used for merits such as smaller key size and faster key computation. As a result, mobile devices are inclined to be E-CC-based cryptosystems rather than traditional cryptosystems. E-CC needs to keep the certificate of the public users and thus increases the storage capacity of the public key infrastructure like the other public key cryptosystems. To address the issue of storage capacity, Shamir [5] proposed an identity-based public key cryptosystem to reduce the barrier of certification management, although that system was not practically oriented. To make the security system practical, Boneh and Franklin [6] proposed an identity-based encryption model using Weil pairing that was adopted using E-CC in 2001.

Sui et al. [7] proposed an improved version of the AKA protocol in 2005 for wireless communication devices, although that scheme failed to withstand attacks of offline password guessing. Determining the solution for offline password-guessing attacks, Liao et al. [8] enhanced the AKA protocol of Sui et al. in 2009. Lu et al. [9] remarked that the enhanced scheme of Liao et al. [8] could not resist parallel guessing attacks. Moreover, Chang et al. [10] proposed a newer version of the AKA protocol to counteract parallel guessing attacks, but that version failed to offer mutual authenticity to users. Kılınç et al. [11] introduced the key-ephemeral strategy for the purpose of attack resiliencies, like replay, key-impersonation, known-key, ephemeral-key, and forward-secrecy, although the authors failed to provide mutual authenticity and thus did not offer a counteracting strategy reliably to server-client systems. Zhang et al. [12] presented a secure authentication scheme for server-client authentication, but their scheme failed to offer services like key-impersonation, server-spoofing, and denial of service (DoS). Thus, we decide to propose a mutual authenticated session key (MASK) that mutually shares the authentication key to enhance the security for multimedia server-client systems. In addition, we analyze the proposed protocol of MASK and compare it with the existing protocols such as those of Lu et al. [9], Chang et al. [10], Kılınç et al. [11], and Zhang et al. [12] in the multimedia server-client environment.

Researchers usually verify mutual authenticity with the proposal of an authentication scheme. We decided to deploy a real server (www.openim-score.org/) a real client (www.uctimsclient.-berlios.de/) to examine the AKA schemes like MASK and those of Lu et al. [9], Chang et al. [10], Kılınç et al. [11], and Zhang et al. [12]. Moreover, we examine the schemes in a traffic analyzer tool (www.ntop.org/) to analyze metrics like call setup time, flooding SIP (Session Initiation Protocol) attack detection rate, and signal congestion rate.

Importantly, the real-time multimedia server and client systems are integrated with authenticated related key security mechanisms that were defined as the important specification of 3GPP in [13]. Section 3 will discuss the detailed review of AKA schemes, such as those of Lu et al. [9], Chang et al. [10], Kılınç et al. [11], and Zhang et al. [12].

1.1. Research contributions

The research contributions are as follows:

1. The proposed protocol of MASK meets all the security requirements that are defined in the 3GPP security mechanisms.
2. Importantly, the MASK mechanism inherits the methodical idea of a symmetric key cryptosystem to expand the sharing key preservation in the 4G networks.
3. The MASK mechanism proficiently shares the session key to curtail the computational overhead of the multimedia server-client.
4. The techniques of password predetermination are used to infer the traffic to improve network performance.
5. The strategy of twofold verification rather than hash verification is used to curtail the message delivery cost.
6. MASK is able to withstand attacks like SIP flooding and examination of results is revealed in Section 5.
7. To verify the secured authentication and security strength, the ntop traffic analyzer (www.ntop.org/) is used.
8. The MASK mechanism enriches the communication efficiency of the multimedia server-client.
9. A multimedia server, namely OpenIMSCore (www.openimscore.org/), is deployed on three different Linux platforms to probe the AKA schemes.
10. A multimedia client such as UCTIMS (www.uctimsclient.berlios.de/) is deployed in three different operating systems (Linux Mint, Ubuntu, CentOS) to probe the voice call sessions.
11. The AKA schemes of Lu et al. [9], Chang et al. [10], Kılınç et al. [11], and Zhang et al. [12] and MASK are integrated with the multimedia server-client to analyze the aforesaid metrics.

The remaining sections are organized as follows. Section 2 presents the related work on the AKA protocol. Section 3 reviews the AKA schemes of Lu et al., Chang et al., Kılınç et al., and Zhang et al. Section 4 proposes the MASK for the multimedia server-client. Section 5 provides the results and discussion. Section 6 concludes the research work.

2. Related work

The key agreement (KA) protocol is usually called a primitive version of cryptography. It is employed to construct a secure session key between the server and client. However, the KA protocol without user authentication is not secure against the anomaly-in-the-middle attack. Thus, researchers and technical experts have proposed several authentication mechanisms [9–14] for the purpose of secure user authentication. The AKA protocol is used to offer mutual authentication to the server-client system. The server-client system shares the session key

when it is generated by the server component. To examine the server component, this paper has deployed a physical multimedia server-client for the consideration of 3GPP features, signal congestion, and computational overhead. Since cryptographic operation is necessitated and expensive, the communication system, namely the server-client, should not have considered the computational limitation [15–18].

The AKA protocol is mainly focused on the traditional public key cryptosystem to reduce the computation of low-power devices. It has lately been proposed for the reduction of computational overhead. Until now, none of the AKA protocols have physically been examined for evaluation results and it has moreover left the following examinations, namely the fulfillment of 3GPP features and signal congestion, undone. Jakobsson and Pointcheval [19] proposed two different AKA mechanisms to reduce the computation of mobile devices. Later, Wong and Chan [20] proposed a mutual authentication mechanism to influential servers and low-computing devices. The protocol of Wong and Chan offers mutual authentication to fulfill the security properties of the server-client environment, but it showed its low computation for the client. We thus decided to examine the computation cost of the multimedia server-client environment.

To examine the forward secrecy (perfect), Smart [21] proposed an identity-based authentication mechanism using the Weil pairing system. Subsequently, Shim [22] revealed that the mechanism of Smart does not provide perfect forward secrecy. In addition, Shim exhibited an identity-based authentication mechanism with fewer Weil pairing operations. Later on, several identity-based authentication mechanisms [23–42] were proposed to reduce the computational cost of mobile devices, although the authentication systems are not yet fully suited [11–28,37–42] for low-power computational devices. Thus, the protocol of MASK is proposed to fulfill the current demand of multimedia server-client systems. We also investigate the testing parameters of computation overhead, 3GPP feature, call setup time, SIP flooding attack detection rate, and signal congestion using a physical multimedia server-client system.

Li and Hwang [29] proposed an authentication scheme to use a random nonce instead of a synchronization clock and it was proven to be efficient in terms of less computation cost. Later on, Li et al. [30] and Das [31] demonstrated that the scheme of Li and Hwang failed to provide proper mutual authentication and resist man-in-the-middle attacks. Yoon and Yoo [32] proposed a robust client-server authentication scheme to offer strong user authentication, although Kim et al. [33] pointed out that the scheme of Yoon and Yoo was not resilient to password (offline) guessing attacks. Recently, Li et al. [34] found some security weaknesses of Das [31] and Lee et al. [35], such as session-key agreement and key-impersonation attacks using biometric-based user authentication schemes. As a consequence, none of the existing authentication schemes [9–12,28–42] fulfill the security properties of the AKA protocol and resist most of the potential attacks, such as password guessing, key impersonation, and so on, in the multimedia client-server environment. Most recently, Deebak et al. [36] presented a secure key AKA protocol scheme to satisfy the promising feature of the 3GPP AKA protocol using IP multimedia server-client systems. However, the scheme of Deebak et al. [36] failed to satisfy key factors such as active-attack on corrupted network, server-spoofing attack, privacy, and reduction of message delivering cost.

3. Review of AKA schemes

To ease the reading, significant notations are provided in Table 1. The AKA schemes of Lu et al. [9], Chang et al. [10], Kılınç et al. [11], and Zhang et al. [12] were studied and their flow methodologies are descriptively explained as follows.

Table 1. Notations.

$D_c, D_{s1} D_{s2} R_c \alpha x, y r_a r_b$	Random integers
M_{Client}, M_{Server}	Communication entities
N	Secure large prime number
P	Large prime order N
D	Uniform distribution dictionary size $\|D\|$
T	Password predetermination
H, H_f	One-way secure hash functions
s	Private key of the server
P_s	Public key of the server
$H_1(), H_2()$	Map-to-point function
$\mathrm{ID}_{MClient}$	Identity of multimedia clients
CS_{Auth}	Client server authentication key
P_{CS}	Prime number of client-server
S_{ki}	Generation of $i$thsession key
Sk_{CS}	Shared session key of client-server
Pvt_{u1}	Private key of $User_1$
Pub_{u1}	Public key of $User_1$
U_{id}	Identity of $User_1$
SS_{key}	Shared session key
S_{id}	Server identity
f(.)	One-way hash function
$f^*(.)$	Another hash function used for session Key
U	User (multimedia)
U_{Name}	User name
RE_{alm}	Server realm (domain name)
δ	Key verifier
k_s	Secret key
$\oplus$	Exclusive operator
P_{wd}	User's password

3.1. AKA Scheme of Lu et al [9]

Lu et al. [9] proposed an enhanced version of the AKA protocol, namely ECAKA (elliptic curve authenticated key agreement). The aim of this protocol was to prevent offline guessing attacks. The execution flows of ECAKA are represented in Table 2 and are as follows.

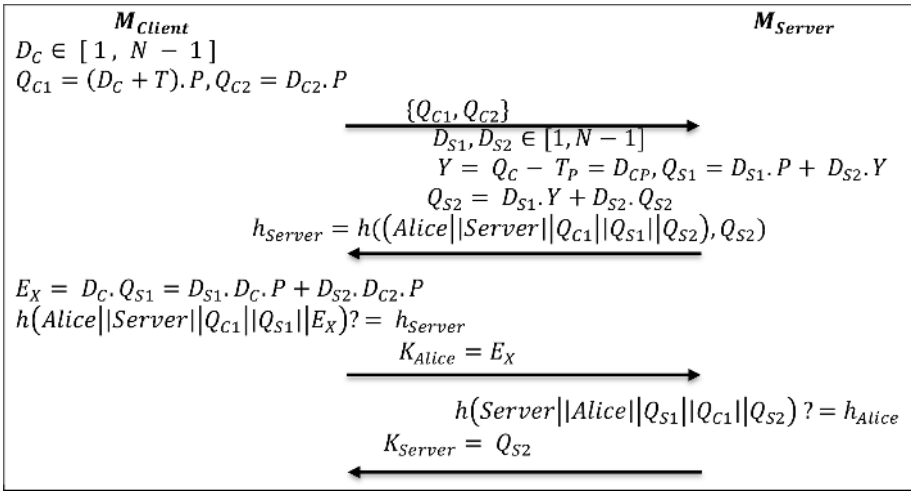
Flow1: First, M_{Client} selects the random number $D_C \in [1, N-1]$ and does the computation of $Q_{C1} = (D_C + T).P, Q_{C2} = D_{C2}.P$. Then M_{Client} sends Q_{C1}, Q_{C2} to Server M_{Server}.

Flow2: Second, M_{Server} selects the two random numbers $D_{S1}, D_{S2} \in [1, N-1]$ and does the computations of $Y = Q_C - T_P = D_{CP}, Q_{S1} = D_{S1}.P + D_{S2}.Y$ and $Q_{S2} = D_{S1}.Y + D_{S2}.Q_{S2}$. Then M_{Server} sends $h_{Server} = h((Alice\,||Server||\,Q_{C1}\,||Q_{S1}||\,Q_{S2})\,, Q_{S2})$ to M_{Client}, and $'h'$ represents a one-way secure hash function and $||\}$ represents the symbol of concatenation.

Flow3: Third, M_{Client} does the computation of $E_X = D_C.Q_{S1} = D_{S1}.D_C.P + D_{S2}.D_{C2}.P$ to validate whether the equality of hash function $h\,(Alice\,||Server||\,Q_{C1}\,||Q_{S1}||\,E_X) = h_{Server}1rver|$ eterminomputationors are failed to offer ng against the attacks, like Spoofing attack of stolen-verifier adheres or not. If

the hash function adheres then M_{Client} does the computation of $h\left(Server\,||Alice||\,Q_{S1}\,||Q_{C1}||\,E_X\right) = h_{Alice}$. Then M_{Client} sends it to M_{Server} with the session key $K_{Alice} = E_X$.

Table 2. AKA scheme of Lu et al.

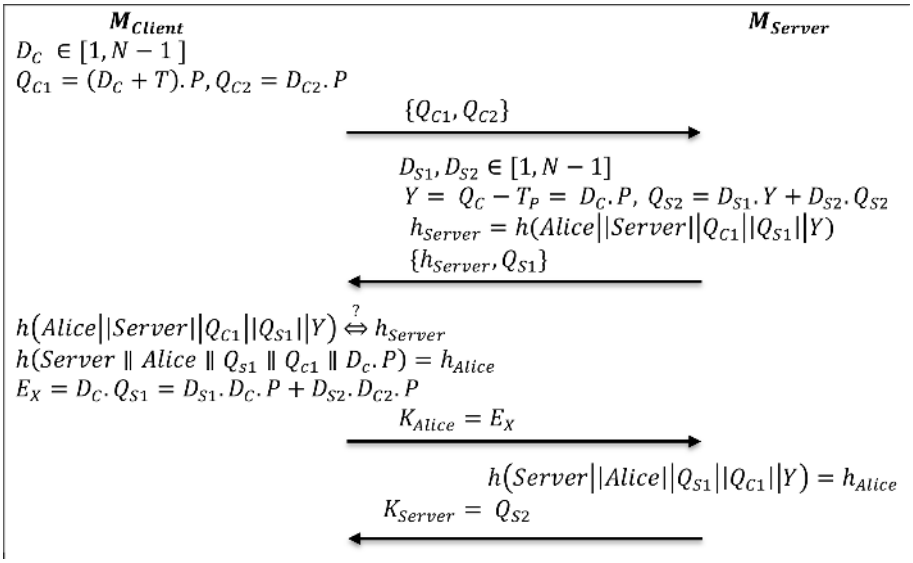


Flow4: Fourth, M_{Server} does the computation of hash function $h\left(Server\,||Alice||\,Q_{S1}\,||Q_{C1}||\,Q_{S2}\right) = h_{Alice}$ if it adheres and then it sets the session key as $K_{Server} = Q_{S2}$.

3.2. AKA scheme of Chang et al [10]

Chang et al. [10] proposed an extension of the authentication scheme of Lu et al., namely EC-PAKA (elliptic curve-based password authenticated key agreement). The aim of this protocol was to prevent password guessing attacks, but it failed to provide mutual authentication for the users. The execution flows of EC-PAKA are represented in Table 3 and are as follows.

Table 3. AKA scheme of Chang et al.



Flow1: First, M_{Client} selects the random number $D_C \in [1, N-1]$ and does the computation of $Q_{C1} = (D_C + T).P, Q_{C2} = D_{C2}.P$. Then M_{Client} sends $Q_{C1}, Q_{C2}\}$ to M_{Server}.

Flow2: Second, M_{Server} selects the two random numbers $D_{S1}, D_{S2} \in [1, N-1]$ and does the computations of $Y = Q_C - T_P = D_C.P, Q_{S2} = D_{S1}.Y + D_{S2}.Q_{S2}$ and $h_{Server} = h(Alice\,||Server||\,Q_{C1}\,||Q_{S1}||\,Y)$. Eventually, M_{Server} sends the message transmission as h_{Server}, Q_{S1} to M_{Client}.

Flow3: On receiving the message transmission of h_{Server}, Q_{S1}, M_{Client} validates whether the equality adheres or not. $h\,(Alice\,||Server||\,Q_{C1}\,||Q_{S1}||\,Y) \overset{?}{\Longleftrightarrow} h_{Server}$.

If it adheres, then M_{Client} does the computation of $h\,(Server \parallel Alice \parallel Q_{s1} \parallel Q_{c1} \parallel D_c.P) = h_{Alice}$ and then M_{Client} sends it to M_{Server} to compute the session key, $E_X = D_C.Q_{S1} = D_{S1}.D_C.P + D_{S2}.D_{C2}.P$ and $K_{Alice} = E_X$.

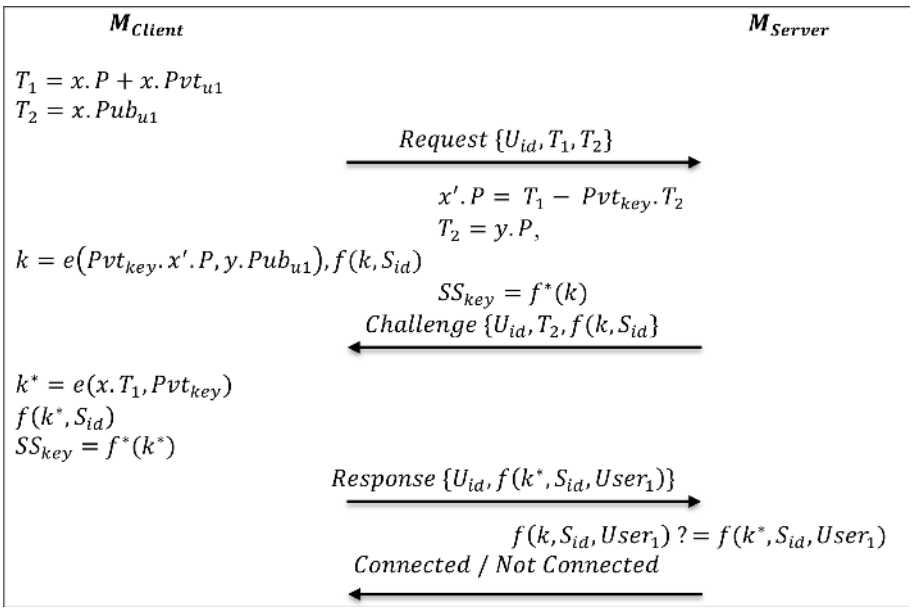
Flow4: On receiving the transmission message of h_{Alice}, M_{Server} validates whether the equality adheres or not. If it adheres, then M_{Server} computes the session key,

$h\,(Server\,||Alice||\,Q_{S1}\,||Q_{C1}||\,Y) = h_{Alice}$ and $K_{Server} = Q_{S2}$.

3.3. AKA Scheme of Kılınç et al [11]

The protocol of Kılınç et al. [11] introduced a strategy of decisional bilinear Diffie–Hellman (DB-DH) and the objective of the strategy is to counteract attacks like replay, key-impersonation, known-key, ephemeral-key, and forward secrecy. Besides, the adversary cannot deduce the session key of the communication parties, since the session key relies on the technical strategy of ephemeral keys, namely x and y. The authentication flows of Kılınç et al. are represented in Table 4 and are as follows:

Table 4. AKA scheme of Kılınç et al.



Flow1: $M_{Client} \rightarrow M_{Server} : Request\{U_{id}, T_1, T_2\}$

Here, a user generates a random integer x and then computes $T_1 = x.P + x.Pvt_{u1}$ and $T_2 = x.Pub_{u1}$ from the $User1$ public and private keys. Eventually, M_{Client} sends a *Request* message U_{id}, T_1, T_2 to M_{Server} with the user identifier of U_{id}.

Flow2: $M_{Server} \rightarrow M_{Client} : Challenge\{U_{id}, T_2, f(k, S_{id})\}$

Upon receiving the $Request\{U_{id}, T_1, T_2$ message from M_{Client}, M_{Server} computes $x^{'}.P = T_1 - Pvt_{key}.T_2$ and then M_{Server} derives a random integer y to compute $T_2 = y.P$, $k = e\left(Pvt_{key}.x^{'}.P, y.Pub_{u1}\right)$ and $f(k, S_{id})$. Then the server sends a user identifier U_{id} as a $Challenge$ message to M_{Client}. The SS_{key} is the shared session key and it is substituted from $SS_{key} = f^*(k)$.

Flow3: $User1 \rightarrow Server : Response\{U_{id}, f\,(k^*, S_{id}, User_1)\}$

Upon receiving the $Challenge\{U_{id}, T_2, f(k, S_{id})\}$ from M_{Server}, M_{Client} derives the key $k^* = e(x.T_1, Pvt_{key})$ from the random integer and private key. Then M_{Client} obtains the hash-value $f\,(k^*, S_{id})$ and compares the former hash-value with $f(k, S_{id})$. If the two hash-values are matched up, then M_{Client} authorizes M_{Server} and sends a $Response$ message to M_{Server}. The SS_{key} is the shared session key and it is substituted from $SS_{key} = f^*(k^*)$.

Flow4: $M_{Server} \rightarrow M_{Client} : Connected/NotConnected$

After receiving the $Response\{U_{id}, f\,(k^*, S_{id}, User_1)$ from M_{Client}, M_{Server} obtains the hash-value $f(k, S_{id}, User_1)$ and then M_{Server} compares the derived hash-value with $f(k^*, S_{id}, User_1)$. If the hash-values are matched up, then M_{Server} authorizes and offers the connection service for M_{Client}.

Though the protocol of Kılınç et al. offers the ephemeral key strategy to counteract against major attacks like replay, key-impersonation, known-key, ephemeral-key, and forward-secrecy, it failed to provide mutual authenticity and thus does not offer a counteracting strategy reliably for the communication parties.

3.4. AKA scheme of Zhang et al [12]

While user M_{Client} wants service access like voice/data from server M_{Server}, M_{Client} and M_{Server} should execute the following flows for the service authentication, as represented in Table 5:

Flow1: $M_{Client} \rightarrow M_{Server} : Request\{U_{Name}, U\}$

M_{Client} generates a random-integer value $r_a \in z_q^*$ to compute $U = r_a.P$, $T_U = r_a.Pub_{key}$ and $U_{Name} = U_{Name} \oplus H(U||T_U)$. Then M_{Client} sends a request message U_{Name}, U to server M_{Server}.

Flow2: $M_{Server} \rightarrow M_{Client}$:$Challenge\{U_{Name}, r_a.P\}$

After receiving the request message U_{Name}, U from M_{Client}, M_{Server} generates a random-integer value $r_b \in z_q^*$ to compute $V = r_b.P$, $T_V = k_s.U$, $SS_{ks} = U.V.P$ and $\omega = H(T_V\,||SS_{ks}||\,V||U)$. Then M_{Server} sends a challenge message RE_{alm}, r_b, ω to M_{Client}.

Flow3: $M_{Client} \rightarrow M_{Server} : Response\{Re_{alm}, \delta\}$

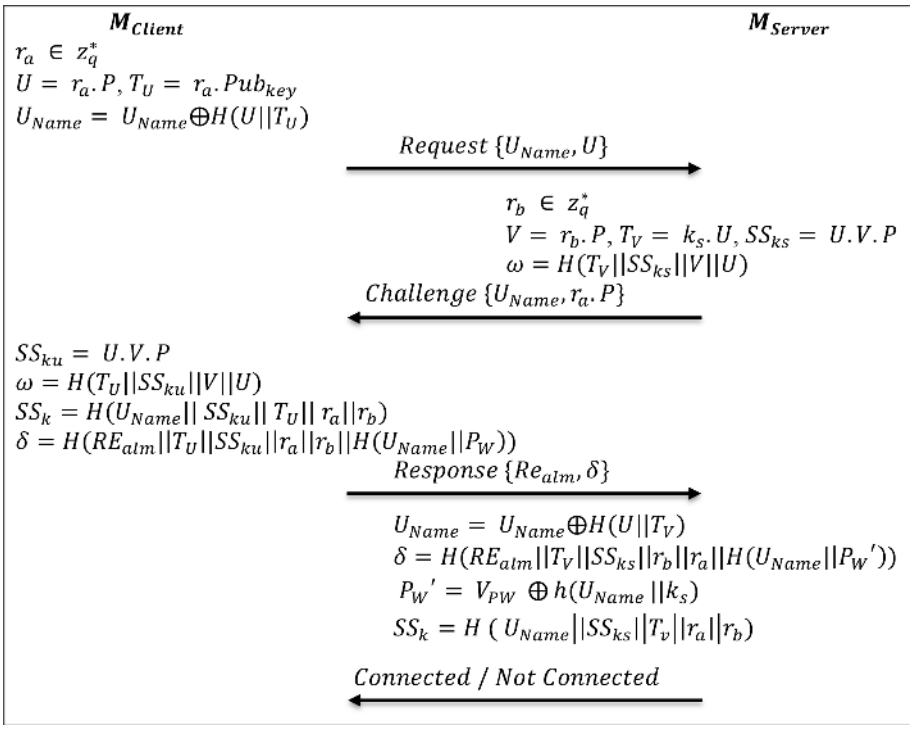
After receiving the challenge message RE_{alm}, r_b, ω, M_{Client} computes $SS_{ku} = U.V.P$ and evaluates whether the expression $\omega = H(T_U|\,|SS_{ku}|\,|V||U)$ holds or not. If the expression fails to hold, then M_{Client} terminates the sessions. Otherwise, M_{Client} computes the shared-session key $SS_k = H(U_{Name}||SS_{ku}||T_U||r_a||r_b)$ and $\delta = H(RE_{alm}|\,|T_U|\,|SS_{ku}|\,|r_a|\,|r_b||H(U_{Name}||P_W))$. Then M_{Client} sends the response message Re_{alm}, δ to M_{Serve}.

Flow4: $M_{Server} \rightarrow M_{Client}$:$Connected/NotConnected$

Upon receiving the response message $\mathbf{Re_{alm}}, \delta$ from M_{Client}, M_{Server} computes $U_{Name} = U_{Name} \oplus H(U||T_V)$ and evaluates whether the expression $\delta = H(RE_{alm}|\,|T_V|\,|SS_{ks}|\,|r_b|\,|r_a||H(U_{Name}||P_{W'}))$. holds or not where $P_W^{'} = V_{PW}Operator[U + 2A01]h(U_{Name}||k_s)$. If it fails to hold, then M_{Server} terminates the

session. Otherwise, M_{Server} shares the session $SS_k = H(U_{Name}||SS_{ks}||T_v||r_a||r_b)$ for the purpose of service authentication.

Table 5. AKA scheme of Zhang et al.



The protocol of Zhang et al. was secure against the attacks of relay, password-guessing, man-in-the-middle, and stolen-verifier, but the protocol failed to offer services like key-impersonation, server-spoofing, and DoS.

4. Proposed authentication scheme of MASK

This section initially presents the MASK using bilinear pairing systems. Then this section compares the 3GPP security properties and computational efficiencies of four AKA schemes, namely those of Lu et al. [9], Chang et al. [10], Kılınç et al. [11], and Zhang et al. [12], with the proposed protocol of MASK. Finally, this section provides a security analysis of the MASK protocol.

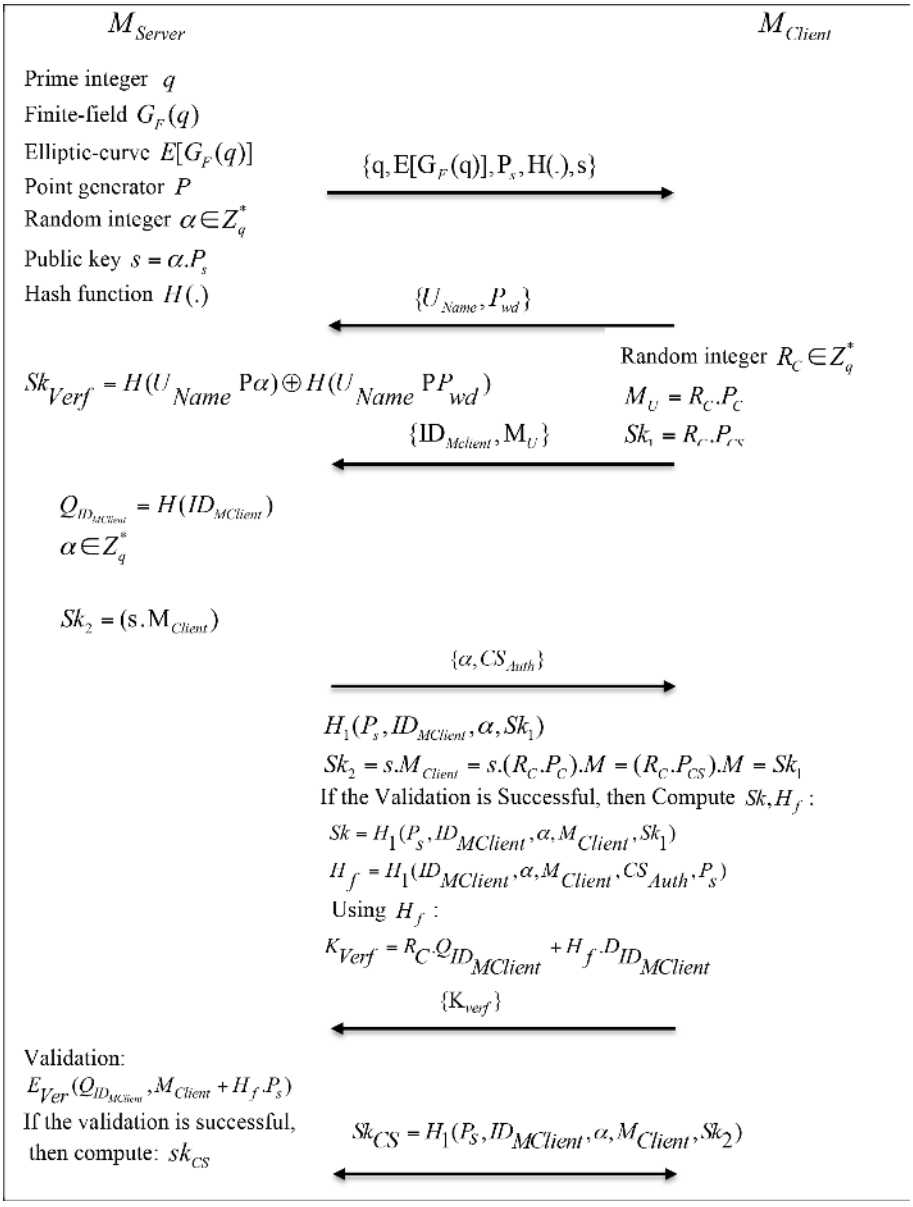
4.1. MASK

This section presents the authenticated communication between two powerful systems, namely a multimedia client (M_{Client}) and multimedia server (M_{Server}). This section will propose the MASK with the E-CC. We present four significant phases, namely system setup, client registration, server-client authentication, and secret-key exchange, for the flow execution of SIP protocol. Table 6 represents the execution flows of the proposed authentication scheme of MASK.

4.2. 1 Phase of system setup

This phase generates the system parameter for the server system M_{Server} and its communication flows are described one by one below.

Table 6. Proposed authentication scheme of MASK.



Flow1: M_{Server} generates a prime integer value q, an elliptic-curve $E(G_F(q))$ over the finite-field $G_F(q)$. P is a point generator of the additive group $E(G_F(q))$.

Flow2: M_{Server} generates a random-integer value $\alpha \in Z_q^*$ as the long-living secret key to compute the long-living public key $s = \alpha.P_s$.

Flow3: M_{Server} chooses a secure hash-value function $H(.)$ and then publishes the server system parameters $q, E(G_F(q)), P_s, H(.), s$.

Phase of client registration

While client M_{Client} wishes to access service like voice/data from server M_{Server}, he/she should enter the client's credentials initially into the client system. The execution flows are as follows:

Flow1: M_{Client} selects the identities, namely U_{Name} and P_{wd} without restriction, and then sends the identities via a secure communication channel. We deploy the secure channel through the Internet and transport layer security protocols.

Flow2: After receiving the identities, like U_{Name} and P_{wd}, M_{Server} computes the secret-key verifier $Sk_{verf} = H(U_{Name}|\alpha) \oplus H(U_{Name}||P_{wd})$ and stores the parameters like U_{Name} and P_{wd} in the home subscriber database.

Phases of server-client authentication and secret-key exchange

The flow methodologies are as follows:

Flow1: M_{Client} selects an integer randomly from $R_C \in Z_q^*$ to compute $M_U = R_C.P_C$ and $Sk_1 = R_C.P_{CS}$. After the above computation, M_{Client} sends $ID_{M_{Client}}, M_U$ to M_{Server}.

Flow2: On receiving the message transmission $ID_{M_{Client}}, M_U$, M_{Server} does the hash computation of $Q_{ID_{MClient}} = H(ID_{M_{Client}})$. Then M_{Server} selects an integer randomly from $\alpha \in Z_q^*$ to compute, $Sk_2 = (s.M_{Client}), CS_{Auth} = H_1(P_S, ID_{M_{Client}}, \alpha, Sk_2)$.

and $H_f = H_1(ID_{M_{Client}}\alpha, M_{Client}, CS_{Auth}, P_S)$. Eventually, M_{Server} sends αCS_{Auth} to M_{Client}

Flow3: On receiving the message transmission αCS_{Auth}, M_{Client} verifies whether CS_{Auth} equals $H_1(P_S, ID_{M_{Client}}, \alpha, Sk_1)$. We use a strategy like $Sk_2 = s.M_{Client} = s.(R_C.P_C).M = (R_C.P_{CS}).M = Sk_1$ to validate the session keys of the multimedia server-client.

If the session key validation is successful, then M_{Client} does the computation of SkandH_f like $Sk = H_1(P_S, ID_{M_{Client}}\alpha, M_{Client}, Sk_1)$ and $H_f = H_1(ID_{M_{Client}}\alpha M_{Client}, CS_{Auth}, P_S)$. Eventually, M_{Client} uses H_f to compute $K_{verf} = R_C Q_{I_{MClient}} + H_f D_{ID_{MClient}}$ and sends it to M_{Server}.

Flow 4: On receiving the message transmission, M_{Server} validates whether $E_{Ver}(Q_{ID_{MClient}}, M_{Client} + H_f.P_S)$ adheres.

If it adheres successfully, then the SC (server-client) computes the common session key $Sk_{CS} = H_1(P_S, ID_{M_{Client}}, \alpha, M_{Client}, Sk_2)$.

The systems, namely M_{Client} and M_{Server} share the common secret key Sk_{CS} to authenticate the communication flows that are as follows:

Flow1: M_{Client} selects a new secret key Sk^{New} to compute $\sigma = H(U_{Name}||Sk||H(U_{Name}||P_{W'}||H(U_{Name}||Sk^{New})$ and then M_{Client} sends the transfer message U_{Name}, σ, P_{wd} to M_{Server}.

Flow2: After receiving the transmission message U_{Name}, σ, P_{wd} from M_{Client}, M_{Server} computes $H(U_{Name}||Sk^{New}) = P_{wd} \oplus H(Sk||H(U_{Name}||P'_{wd}))$ and validates whether the former expression $\sigma = H(U_{Name}||Sk||H(U_{Name}|P'_{wd}))||H(U_{Name}||Sk^{New})$ holds or not where $P'_{wd} = P_{wd} \oplus H(U_{Name}||\alpha)$. If the expression does not hold, then M_{Server} terminates the sessions with M_{Client}. Otherwise, M_{Server} modifies P_{wd} with $Sk^{New} = H(U_{Name}|[ERR : md : MbegChr = 0x007C, MendChr = 0x0029, nParams = 1] \oplus H(U_{Name}||P'_{wd})$.

The above steps are run to share the common session key between the multimedia client $MClient$ and server $MServer$ securely. The common session key is shared to offer mutual authentication, security privacy, and preservation consistently.

4.3. 3GPP security features: a comparison

Table 7 illustrates the comparison of 3GPP security properties with the AKA protocols. The proposed mechanism of MASK is able to achieve inclusive performance compared to the existing schemes of Lu et al.,

Chang et al., Kılınç et al., and Zhang et al. The proposed protocol of MASK endeavors to:

Table 7. Comparison of 3GPP security properties with the AKA protocols.

	Lu et al. [9]	Chang et al. [10]	Kılınç et al. [11]	Zhang et al. [12]	MASK
D1	Symmetric	Symmetric	Symmetric	Symmetric	Symmetric, with the integral technique of 'T' and 's' and D-H
D2	No	No	Partial	Partial	Yes
D3	No	Partial	Not reliable	Not reliable	Yes (reliable)
D4	No	No	No	No	Yes
D5	Partial	No	No	No	Yes
D6	No	No	No	No	Yes
D7	No	No	No	No	Yes

D1- Adhere to the type of cryptosystem.

D2- Adhere to counteracting attacks, like replay, redirection, active-corrupted network, known key-secure, key compromise-impersonate, man-in-the-middle, password-guessing, server-spoofing, stolen-verifier, DoS, and unknown key-share.

D3- Adhere to share the key mutually between the multimedia server-client to secure the communication.

D4- Adhere to using the predetermination key to reduce the computational overhead.

D5- Adhere to curtailing the signal congestion by the strategy of password predetermination.

D6- Adhere to securing all the components of the multimedia server.

D7- Adhere to mutual authenticity, privacy preservation, perfect forward secrecy, unknown-key share, known key-share, and reduction of message delivery cost.

5. Computational efficiency of AKA schemes

Table 8 compares the computational efficiencies of MASK with the other four existing schemes. In the scheme of Lu et al., the client has to execute 3 scalar-multiplications and 2 hash-operations while the server has to execute 5 scalar-multiplications, 3 point-additions, and 2 hash-operations. In the scheme of Chang et al., the client has to execute 7 scalar-multiplications, 2 point-additions, and 4 hash-operations while the server has to execute 6 scalar-multiplications, 2 point-additions, and 4 hash-operations. In the scheme of Kılınç et al., the client has to execute 3 scalar-multiplications, 1 point-addition, 1 bilinear pairing, and 3 hash-operations while the server has to execute 4 scalar-multiplications, 1 point-subtraction, 1 bilinear pairing, and 3 hash-operations. In the scheme of Zhang et al., the client has to execute 3 scalar-multiplications, 4 hash-operations, and 1 scalar-exponentiation while the server has to execute 3 scalar-multiplications, 4 hash-operations, and 1 scalar-exponentiation. In MASK, the client has to execute 4 scalar-multiplications, 1 point-addition, and 3 hash-operations, whereas the server has to execute 1 scalar-multiplication, 1 point-addition, and 4 hash-operations. To have enough strength of security, 160-bit key size is initialized for the four previous AKA protocols and MASK.

Table 8. Computational efficiencies of AKA protocols

Parameters	MASK		Lu et al. [9]		Chang et al. [10]		Kılınç et al. [11]		Zhang et al. [12]	
	C	S	C	S	C	S	C	S	C	S
Scalar-multiplication	4	1	3	5	7	6	3	4	3	3
Point-subtraction	0	0	0	0	0	0	0	1	0	0
Point-addition	1	1	0	3	2	2	1	0	0	0
Bilinear pairing	0	0	0	0	0	0	1	1	0	0
Hash-operation	3	4	2	2	4	4	3	3	4	4
Scalar exponentiation	0	0	0	0	0	0	0	0	1	1
Key size	160 bits									
C – client S – server										

6. Security analysis of MASK

This section shows that the proposed protocol of MASK can mutually authenticate the multimedia server-client using the session key sharing mechanism to avoid SIP flooding attacks. The MASK can also provide perfect forward secrecy for the multimedia server-client to resist offline password guessing attacks.

Known key-secure: Assume an adversary with a previous session key of 'S_{kCS}' shared as a common session key for the communication parties, namely Alice and Bob. To deduce the common session key, the adversary should be able to verify the computed session key $S_{k_1} = S_{k_2}$. Since the verification is hard for the adversary, we thus assert that the protocol of MASK counteracts the attack of known key.

Key compromise-impersonation: Suppose the client (Alice/Bob) makes the adversaries aware of the session key. Though the adversaries possess the session key of the client, the adversaries cannot do session-key verification without the parameter of P_{CS} (client-server prime number). Hence, the protocol of MASK has a feature of resilience for key compromise-impersonation to counteract against key-impersonation attacks.

Unknown-key share: Since the protocol of MASK does not support the precondition/selection of session keys, the adversaries cannot determine the actual session keys of the communication parties. Thus, the protocol of MASK can counteract attacks of unknown-key share.

Redirection attack: Suppose an adversary has a device that is simulated to invoke the functions of the multimedia server and client. Thus, the adversary can forge the messages of legitimate clients on the networks. To resolve this issue, the protocol of MASK has discovered a common session for the multimedia communication parties. Hence, the protocol of MASK can counteract attacks of redirection.

Active attack on corrupted network: Assume a network is completely corrupted, and thus the adversaries can deduce the session keys of communication parties to impersonate a legal network to connect with the client. To prevent this, the protocol of MASK has invoked a common session key verification and thus the scenario of illegitimate networks does not exist. In addition, the session keys of communication parties are recorded in the database of the subscriber server and thus the server of SCSCF makes usage unavailable for illegitimate networks.

Mutual authentication: The initial message of the multimedia client contains the challenge number that is used to be encrypted with the cipher key to be shared later by the multimedia client and SCSCF (serving call session control function). To validate the shared key, the SCSCF would receive and decrypt the message into the original text. If it is decrypted successfully into the original text, then it proves that the shared keys are authenticated by the multimedia client and SCSCF. Even if any attackers/intruders steal the shared key of the multimedia client or SCSCF, the multimedia client/SCSCF can deduce/verify in the second round-trip

of message transmission. This is owing to the parameter of 's' in the Diffie–Hellman problem to compute the session key from the one-way hash function. Most importantly, the parameter of P_s is already shared with the multimedia server-client to show the authenticated session key reliability.

Perfect forward secrecy: Even though the cipher key and challenging number are known to the adversary, they cannot compute the session key for the multimedia server-client. This is owing to the parameters of R_c and R_s that are to be determined from the random number belonging to Z_{q*}. To determine the shared session key of the multimedia server-client, the attackers should have to guess the correct one-way hash function. This is usually very hard. Thus, the protocol of MASK adheres to the property of perfect forward secrecy.

Password-guessing (online) attack: If any adversary wants to presume the secret key of a legitimate user for the logon server, he/she must contrive a rational secret key P_s, but the adversary cannot formulate a valid secret key without the knowledge of $H_f = H_1(ID_{M_{Client}} \alpha M_{Client}, CS_{Auth}, P_S)$ and thus the protocol of MASK can withstand attacks of password guessing (online).

Stolen-verifier attack: Assume that the client credential is breached and thus the adversary may use the breaching information to steal the session keys of the communication parties. Though the adversary has the breaching information of the client, the adversary cannot invoke the parameter CS_{Auth} that is used to be computed while the session keys are being shared between the communication parties. Thus, we assert that the protocol of MASK can withstand attacks of stolen verifiers.

Man-in-the-middle attack: Assuming that an adversary wants to carry out the attack of man-in-the-middle, he/she must secretly listen the logon request/response message, message communication, and session key sharing between the communication parties, but the adversary cannot invoke the parameters, namely $S_k, S_{k1}, S_{k2}, P_S, \text{and} \alpha$. Thus, we assert that the MASK protocol can withstand attacks of man-in-the-middle.

Server-spoofing attack: Assume a mischievous server $M_{Server1}$ wants to betray M_{Client} in lieu of M_{Server} and the objective of server mischief is to invoke the session key of M_{Server}, although the protocol of MASK cannot render the session key without the successful verification. Thus, we assert that the protocol of MASK can withstand attacks of server-spoofing.

Replay attack: In the protocol of MASK, the random numbers $r_a \text{and} r_b$ are selected randomly to let the adversaries out of the systems. Since the random values are different for every authentication, the adversary cannot counterfeit the procedural steps of authentication. Besides, the off-sync feature will earn the authentication failure for the illegitimate user and thus the protocol of MASK withstands attacks of replay.

Denial of service attack: Since a mischievous client launches the attack through the SCSCF, the authentication protocol of MASK has a strategy of detection in the HSS as K_{verf}. As the protocol of MASK generates its common session key using its knowledgeable parameters, like $r_a r_b \text{RCandPC}$, thus the MASK protocol can verify its generated session key through the HSS to withstand attacks of DoS.

Providing privacy for multimedia users: The entities of multimedia, namely the client and server, use two-party key-transfer authentication protocol (K-TAP)/two-party key-agreement authentication protocol (K-AAP) [37] to plot the route procedure between the multimedia entities. For the KA protocol of the IP Multimedia Subsystem (IMS), the private identity of the multimedia client is removed from the original SIP (session initiation protocol) transmission. The message of SIP is routed through the call session control function (CSCF) using the domain of IMS to unveil the public identity of the multimedia client. Thus, we assert that the protocol of MASK provides privacy preservation for multimedia server-client systems.

Reducing message delivery cost: Media streaming is protected using the key sharing protocol with reasonable usage of latency, although the solution of MIKEY [11] uses a strategy of session description protocol

(SDP) to minimize the additional message delivery cost. To mitigate the message delivery cost, the server-client systems employ the strategy of twofold verification rather than hash verification. The mitigation of message delivery cost also offers the minimum signal congestion for the multimedia server-client system.

Mutual authentication: This strategy is commonly employed to mitigate the spam over IP-telephony (SPIT) [12]. In addition, it ensures the validity of the multimedia server-client or service provider of the multimedia domain network. The protocol of MASK uses the parameters K_{verf}andE_{Ver} to offer twofold verification strategy to authenticate the sessions of server-client systems. We thus assert that the protocol of MASK holds the property of AKA protocol.

The next section will demonstrate the multimedia server-client setup for the AKA schemes of Lu et al., Chang et al., Kılınç et al., and Zhang et al. and MASK. Then we will analyze metrics like call setup time and flooding (SIP) attack detection rate through a real-time multimedia system to show the importance of the MASK protocol.

7. Results and discussion

The multimedia server-client is installed under three Linux operating systems (OSs). The OS offers a Pentium i5-4440 processor and it is capable of 3.10 GHz clock speed, 6.0 MB cache, and DDR3-1333/1600 memory type. The cryptographic library of MIRACL (http://www.shamus.ie/index.php) is installed and configured in Linux OS. It is enabled in the environment of the multimedia server-client to provide functions like multiprecision rational arithmetic integers.

To examine the voice service realistically, we deploy five multimedia servers of OpenIMSCore (http://www.openim-score.org/) in Linux PCs (that is, Linux Mint) with unique IP addresses and domain names. The IP address and domain name of server1 are {192.68.77.30,test1.test}, whereas those of server2 are {192.168.77.31,test2.test}, those of server3 are {192.168.77.32,test3.test}, those of server4 are {192.168.77.33, test4.test}, and those of server5 are {192.168.77.34,test5.test}. To play the voice service physically between the laptops/desktops, we install ten multimedia clients of UCTIMS (http://uctimsclient.berlios.de/) in Linux PCs (that is, Linux Mint) to establish the service of voice calls through the multimedia server to examine the metrics of call setup time and flooding (SIP) attack detection rate.

The physical multimedia server-client environment is depicted in Figure 1. The servers of multimedia are composed of three different CSCFs, namely proxy, serving, and interrogating, to process the signaling packets of SIP and one user database server, namely the home subscriber server (HSS). The AKA protocol of MASK is integrated with $M_{Server1}$, whereas the protocols of Lu et al., Chang et al., Kılınç et al., and Zhang et al. are integrated with $M_{Server2}$, $M_{Server3}$, $M_{Server4}$, and $M_{Server5}$to cross-examine the metrics of call setup time, flooding (SIP) attack detection rate, and signal congestion rate. The voice call is established between the multimedia server-client through the Internet service of either WiFi or WLAN to examine the former metrics.

The multimedia servers of $M_{server1}$, $M_{server2}$, $M_{Server3}$, $M_{Server4}$, and $M_{Server5}$are run in parallel for 4 h. The initial 2 h are used to probe the call setup time (voice service) and the remaining 2 h are used to examine the flooding attack detection rate. To inspect the flooding detection rate genuinely, we install and configure the flooding tools of SIP and the resources of codes are taken from http://www.backtrack-linux.org/wiki/index.php/Pentesting_ VOIP. The network traffic tool of ntop (www.ntop.org/) is installed and configured with the multimedia server-client systems to analyze the metrics logically. The voice codec of G.723 is configured with the multimedia client for better exchange of transmission rate. The clients are configured

physically with the network of IEEE 802.11a. The SIP flooding attacks like invite, reinvite, and rtp (real-time transport protocol) are used to test the true detection rate over the false detection rate.

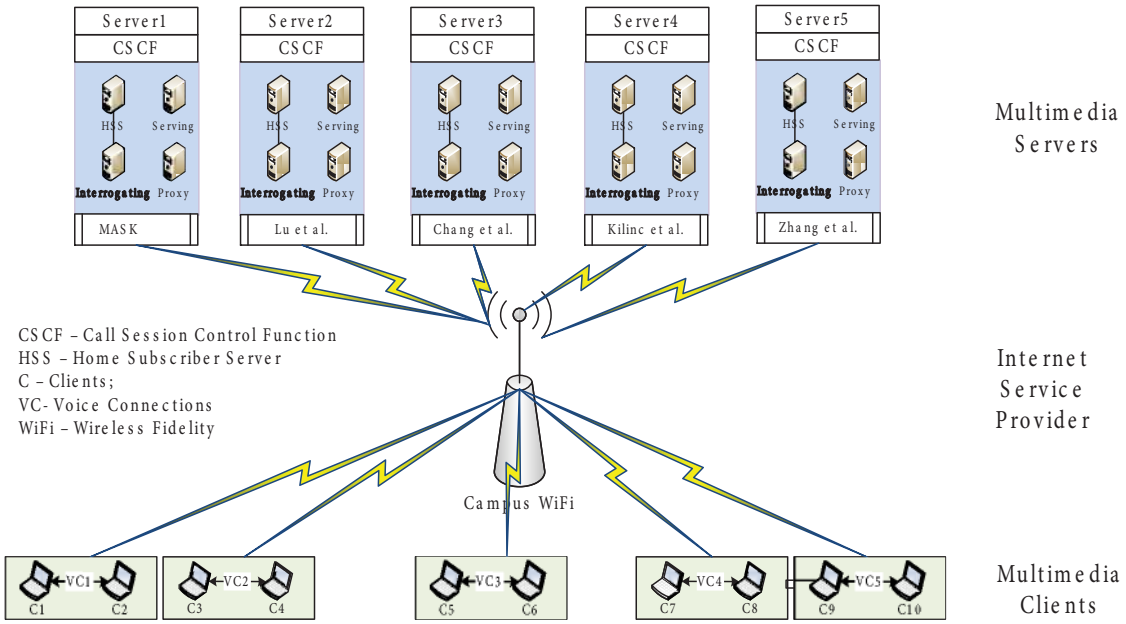


Figure 1. Multimedia server-client environment.

The forthcoming sections will demonstrate the metrics such as call setup time, flooding (SIP) detection rate, and signal congestion rate in the environment of the multimedia server-client.

Figure 2 illustrates call setup time. The multimedia server is run in five Linux platforms through which the multimedia clients, namely client 1-2, client 3-4, client 5-6, client 7-8, and client 9-10, are established with voice call service to probe the call response time every 40 min. Since the proposed MASK protocol uses twofold verification, namely $K_{verf} and E_{Ver}$, to establish the service, Server1-Client1-2 with MASK shows the minimum response time compared to Server2-Client3-4 with Lu et al., Server3-Client5-6 with Chang et al., Server4-Client7-8 with Kılınç et al., and Server5-Client9-10 with Zhang et al. Most importantly, Server1-Client1-2 with MASK regularly initiates the voice call at around 0.231 s and 0.261 s, whereas Server2-Client3-4 with Lu et al. establishes the voice call at around 0.388 s, Server3-Client5-6 with Chang et al. establishes the voice call at around 0.452 s, Server4-Client7-8 with Kılınç et al. establishes the voice call at around 0.491 s, and Server5-Client9-10 with Zhang et al. establishes the voice call at around 0.371 s.

Figure 3 illustrates the flooding (SIP) attack detection rate. The adversary has the breaching information of the client system, but he/she cannot invoke the parameter CS_{Auth} to compute the session keys. So as to examine the SIP flooding attack realistically, the flooding tools of invite, reinvite, and bye (http://www.backtrack-linux.org/wiki/index.php/Pentesting_ VOIP) are installed and configured with the multimedia client system. When we inspected the 'SIP Traffic' after the attacks being triggered, it was shown that Server1-Client1-2 with MASK achieves the acceptable true detection rate (close to 93.5%) when its false positive rate is even set to 3%, whereas the other security mechanisms like Server2-Client3-4 with Lu et al., Server3-Client5-6 with Chang et al., Server4-Client7-8 with Kılınç et al., and Server5-Client9-10 with Zhang et al. achieve much lower true detection rates (close to 43.1%, 42.1%, 44.1%, and 45.1%) when their false positive rate is even set to 1.5%.

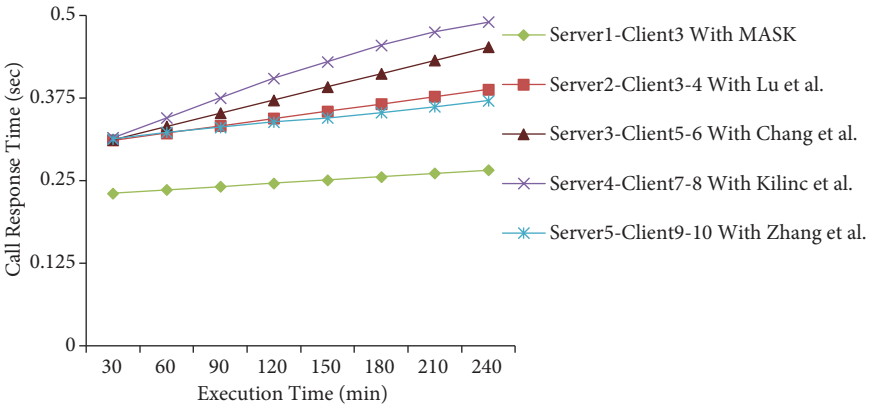


Figure 2. Call setup time.

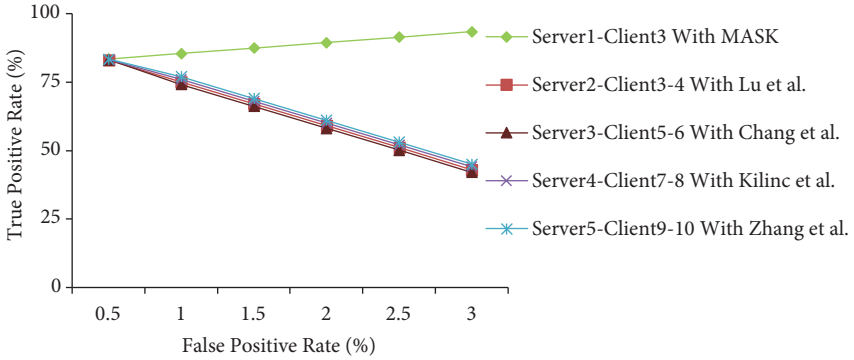


Figure 3. Flooding (SIP) attack detection rate.

The proposed mechanism of MASK has retracted the on-time computation of the authentication key by the strategic technique of key predetermination ('T'). The parameter of the server private key ('s') is used to curtail the pairing computation of the multimedia server-client. The former mechanism helps to ease the computational time of the server-client authentication and the latter mechanism is employed to minimize the traffic congestion of the multimedia server-client. Figure 4 illustrates that Server1-Client1-2 with MASK has much less signal congestion in comparison with the existing schemes, namely Server2-Client3-4 with Lu et al., Server3-Client5-6 with Chang et al., Server4-Client7-8 with Kılınç et al., and Server5-Client9-10 with Zhang et al. The results for signal congestion were validated through the traffic analyzer tool of ntop.

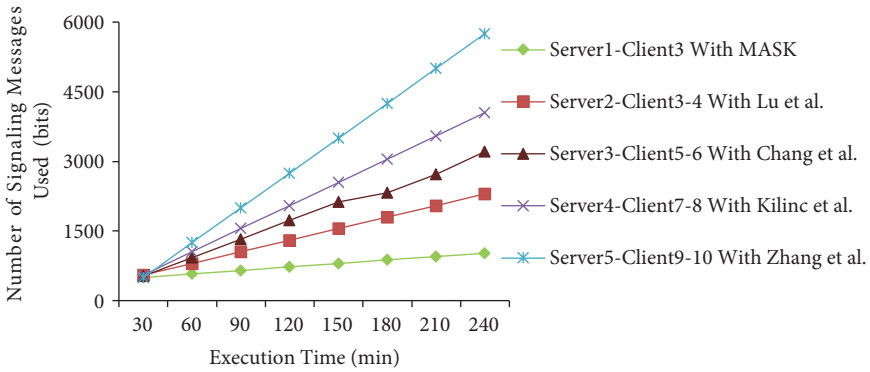


Figure 4. Signal congestion rate.

When we analyzed the real-time multimedia server-client systems, it was verified that the proposed mechanism of MASK can incur better session key security, and thereby the voice call establishment of the server-client system abides by all the security-associated mechanisms of 3GPP. Furthermore, the MASK strategy is well suited for the protection of the media system against SIP flooding attacks.

8. Conclusion

Since the existing protocols like those of Lu et al. [9], Chang et al. [10], Kılınç et al. [11], and Zhang et al. [12] have not had salient 3GPP features of reasonable computational overhead, mutual authenticity, and signal congestion, we have thus proposed the mechanism of MASK for real-time multimedia server-client systems. The proposed authentication mechanism of MASK can mitigate the computational overhead comparatively better than the existing protocols. Besides, the proposed mechanism of MASK satisfies all security features of the 3GPP AKA protocol, such as mutual authentication, forward secrecy, privacy, known-key security, and so on. Importantly, the MASK exploits the idea of a symmetric key cryptosystem to achieve the feature of key preservation in 4G networks.

Moreover, it skillfully shares the session key to ease the computational overhead of the multimedia server-client systems. The technique of twofold verification is used to reduce the message delivery cost. The experimental results of the multimedia server-client system show that the proposed mechanism of MASK can mitigate call setup time, flooding attack detection rate, and signal congestion rate relatively better than the methods of Lu et al., Chang et al., Kılınç et al., and Zhang et al. Above all, the mechanism of MASK meets the 3GPP specifications for end-to-end security improvement.

Acknowledgment

The corresponding author would like to thank TATA Consultancy Services for research guidance and financial support under the scheme of the Research Scholar Program.

References

[1] Camarillo G, Garcia Martin MA. The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds. 2nd ed. New York, NY, USA: Wiley, 2006.

[2] Third Generation Partnership Project. Technical Specification Group Services and System Aspects: 3G Security and Access Security for IP-Based Services. 3GPP TS 33.203 2008; V7.9.0.

[3] Third Generation Partnership Project. Technical Specification Group Services and System Aspects: 3G Security Network Domain Security IP Network Layer Security. 3GPP TS 33.210 2010; V6.6.0.

[4] Almasalha F, Agarwal N, Khokhar A. Secure multimedia transmission over RTP. In: Tenth IEEE International Symposium on Multimedia; 15–17 December 2008; Berkeley, CA, USA. pp. 404-411.

[5] Shamir A. Identity-based cryptosystems and signature schemes. In: Advances in Cryptology - CRYPTO 84; 19–22 August 1984; Santa Barbara, CA, USA. pp. 47-53.

[6] Boneh D, Franklin M. Identity based encryption from the Weil pairing. SIAM J Comput 2003; 32: 586-615.

[7] Sui A, Hui L, Yiu S, Chow K, Tsang W, Chong C, Pun K, Chan H. An improved authenticated key agreement protocol with perfect forward secrecy for wireless mobile communication. In: 2005 IEEE Wireless Communications and Networking Conference; 13–17 March 2005; New Orleans, LA, USA. pp. 2088-2093.

[8] Liao YP, Wang SS. A secure dynamic ID based remote user authentication scheme for multi-server environment. Comput Stand Inter 2009; 31: 24-29.

[9] Lu R, Cao Z, Zhu H. An enhanced authenticated key agreement protocol for wireless mobile communication. Comput Stand Inter 2007; 29: 647-652.

[10] Chang CC, Chang SC. An improved authentication key agreement protocol based on elliptic curve for wireless mobile networks. In: 2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing; 15–17 August 2008; Harbin, China. pp. 1375-1378.

[11] Kılınç HH, Allaberdiyev Y, Yanık T, Erdem SS. Efficient ID based authentication and key agreement protocols for the session initiation protocol. Turk J Electr Eng Co 2015; 23: 560-579.

[12] Zhang Z, Qi Q, Kumar N, Chilamkurti N, Jeong HY. A secure authentication scheme with anonymity for session initiation protocol using elliptic curve cryptography. Multimed Tools Appl 2015; 74: 3477-3488.

[13] Third Generation Partnership Project. Technical Specification Group Services and System Aspects: 3G Security Formal Analysis of the 3G Authentication Protocol. 3GPP TR 33.902 1999; V3.1.0.

[14] Diffie W, Hellman M. New directions in cryptography. IEEE T Inform Theory 1976; 22: 644-654.

[15] ElGamal T. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE T Inform Theory 1985; 31: 469-472.

[16] Rivest R, Shamir A, Adelman L. A method for obtaining digital signature and public key cryptosystem. Commun ACM 1978; 21: 120-126.

[17] Tseng YM, Wu TY, Wu JD. A pairing-based user authentication scheme for wireless clients with smart cards. Informatica 2008; 19: 285-302.

[18] Nam J, Lee J, Kim S, Won D. DDH-based group key agreement in a mobile environment. J Syst Software 2005; 78: 73-83.

[19] Tseng YM. GPRS/UMTS-aided authentication protocol for wireless LANs. IEE P-Commun 2006; 153: 810-817.

[20] Tseng YM. A resource-constrained group key agreement protocol for imbalance wireless networks. Comput Secur 2007; 26: 331-337.

[21] Jakobsson M, Pointcheval D. Mutual authentication for low-power mobile devices. In: Fifth International Conference on Financial Cryptography; 19–22 February 2001; Grand Cayman, British West Indies. pp. 178-195.

[22] Wong DS, Chan AH. Efficient and mutually authenticated key exchange for low power computing devices. In: Advances in Cryptology - ASIACRYPT'01; 9–13 December 2001; Gold Coast, Australia. pp. 172-289.

[23] Smart NP. An identity based authenticated key agreement protocol based on the Weil pairing. Electron Lett 2002; 38: 630-632.

[24] Shim K. Efficient ID-based authenticated key agreement protocol based on the Weil pairing. Electron Lett 2003; 39: 653-654.

[25] Chen L, Cheng Z, Smart NP. Identity-based key agreement protocols from pairings. Int J Inf Secur 2007; 6: 213-241.

[26] Chen L, Kudla C. Identity-based authenticated key agreement from pairings. In: 16th IEEE Computer Security Foundation Workshop; 30 June–2 July 2003; Pacific Grove, CA, USA. pp. 219-233.

[27] Choie YJ, Jeong E, Lee E. Efficient identity-based authenticated key agreement protocol from pairings. Appl Math Comput 2005; 162: 179-188.

[28] Wang S, Cao Z, Bao H. Two-pass ID-based authenticated key agreement protocol with key confirmation using pairings. In: First International Multi-Symposiums on Computer and Computational Sciences; 20–24 June 2006; Hangzhou, China. pp. 109-112.

[29] Li CT, Hwang MS. An efficient biometrics-based remote user authentication scheme using smart cards. J Netw Comput Appl 2010; 33: 1-5.

[30] Li X, Niu JW, Ma J, Wang WD, Liu CL. Cryptanalysis and improvement of a biometric-based remote authentication scheme using smart cards. J Netw Comput Appl 2011; 34: 73-79.

[31] Das AK. Analysis and improvement on an eficient biometric-based remote user authentication scheme using smart cards. IET Inform Secur 2011; 5: 145-151.

[32] Yoon EJ, Yoo KY. Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem. J Supercomput 2013; 63: 235-255.

[33] Kim HH, Jeon WR, Lee KW, Lee YH, Won DH. Cryptanalysis and improvement of a biometrics-based multi-server authentication with key agreement scheme. In: 2nd International Conference on Computational Science and Its Applications; 18–21 June 2012; Salvador de Bahia, Brazil. pp. 391-406.

[34] Li X, Niu JW, Wang ZB, Chen C. Applying biometrics to design three factor remote user authentication scheme with key agreement. Secur Commun Netw 2013; 7: 1488-1497.

[35] Lee CC, Hsu CW. A secure biometric-based remote user authentication with key agreement scheme using extended chaotic maps. Nonlinear Dynam 2013; 71: 201-211.

[36] Deebak BD, Muthaiah R, Thenmozhi K, Swaminathan PI. Analyzing secure key authentication and key agreement protocol for promising features of IP multimedia subsystem using IP multimedia server-client systems. Multimed Tools Appl 2016; 75: 2111-2143.

[37] Arkko J, Carrara E, Lindholm F, Naslund M, Norrman K. MIKEY: Multimedia Internet KEYing. Internet Engineering Task Force 2004; RFC 3830.

[38] Wang S, Cao Z, Cao F. Efficient identity-based authenticated key agreement protocol with PKG forward secrecy. Int J Netw Secur 2008; 7: 181-186.

[39] Wang S, Cao Z, Choo KK. Provably secure identity-based authenticated key agreement protocols without random oracles. Cryptology ePrint Archive, Report 2006/252, 2006, available at https://eprint.iacr.org/2006/446.pdf.

[40] Wang S, Cao Z, Dong X. Provably secure identity based authenticated key agreement protocols in the standard model. Chinese J Comput 2007; 30: 1842-1854.

[41] Tian HB, Susilo W, Ming Y, Wang YM. A provable secure ID-based explicit authenticated key agreement protocol without random oracles. J Comput Sci Technol 2008; 23: 832-842.

[42] Yeh HT, Sun HM. Password-based user authentication and key distribution protocols for client–server applications. J Syst Software 2004; 72: 91-103.