DS-AKA: Dynamic and Secure Authentication and Key Agreement Protocol for UMTS Networks

Mustafa A. AL-Fayoumi

Abstract—The authentication and key agreement (AKA) protocol of Universal Mobile Telecommunication System (UMTS) adopts the security features of Global System for Mobile (GSM) in order to interwork with GSM compatibility. Furthermore, the UMTS increases more security features than GSM to design an authentication and key agreement protocol, which is called UMTS AKA protocol. The UMTS AKA is still vulnerable to redirection and man-in-the-middle attacks, which allow an adversary to redirect user traffic form a network to another and eavesdrop or mischarge the subscribers in the system. Moreover, UMTS AKA protocol has performance problems, including bandwidth consumption between a serving network and user's home network and space overhead of the serving network. In this paper, by using a key hash chaining authentication technique an innovative contribution introduces a dynamic and secure AKA protocol, called DS-AKA to resolve the security issue and cope with performance problems. A security analysis and comparison with related work shows that DS-AKA protocol is more secure and the network can be operated in a more efficient way.

Index Terms— Authentication Protocol, Key agreement protocol, Universal Mobile Telecommunications Systems (UMTS), Security, Mobile Station, and Authentication Vector, Wireles communication.

1 INTRODUCTION

With the advancement of wireless communication and computer technologies, mobile communication provides more versatile, portable and affordable networks than ever [1], [2]. Therefore, the number of users of mobile communication networks has increased rapidly. The convenience of communication not only brings a new set of technical problems, but also raises a new class of interesting applications. This is due to the change in communication from single-medium oriented into multimedia communication such as image, computing data, internet services, e-commerce [3], and so on. At the same time, how to decrease the risk of masquerading legal users, and protect privacy on the radio channels are becoming a very important issues [4].

Since the transmission interfaces are over insecure communication channels, security is one of the most important requirements for the exchange of user's or systems' private data. Therefore, precautionary security measures for mobile communication systems should be provided. As a solution to prevent the illegal access of frauds and eavesdroppers, authentication and confidentiality are essential security services to subscribers and the service provider [5]. Take cellular mobile communication systems for example, entities of a cellular mobile communication system. There are three entities participating in the UMTS security architecture. Firstly, a home environment (HE), with which the MS contracts. Secondly, a foreign network, which is called serving network (SN). An MS can connect to an HN or an SN. Thirdly, a mobile station (MS), which is on behalf of a user. Figure 1 illustrates the UMTS architecture [6].

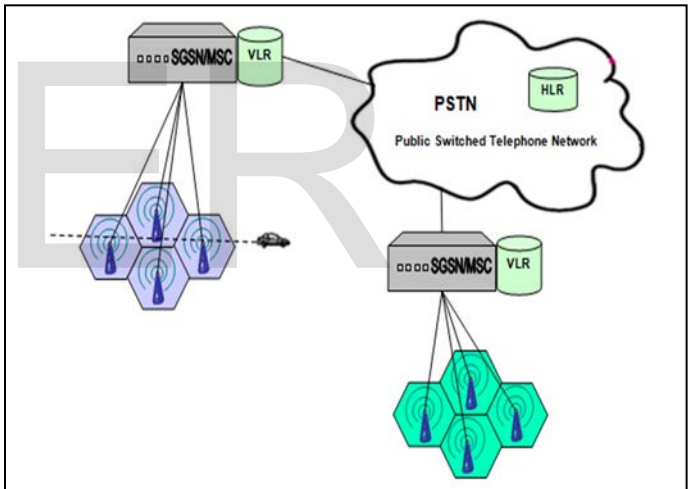

Fig. 1. Mobile network architecture

The GSM was introduced in 1990 by the European Telecommunication Standard Institute [7]. The GSM takes account of security issues and employs some security measures in designing the 2G to provide user authentication and data confidentiality [8]. Improved versions of the GSM protocol were subsequently proposed [9], [10], [11], [12]. However, the security features of the GSM are gradually insufficient for the current demands of the mobile system. A more powerful security mechanism is essential to address and improve current GSM security.

The UMTS AKA authentication protocol holds the framework of the GSM but provides new and significant enhancement features such as mutual authentication, agreement on an integrity key between the user (MS) and the serving network (SN), and assured freshness of agreed cipher key and integrity key. In the UMTS authentication protocol, according to the properties of the Message Authentication Code (MAC), the mobile station and the Home Location Register and Authenti-

• *Mustafa A. Al-Fayoumi, Assistant Professor of computer science. Currently he is the Dean Assistant for Quality and Development at College of Computer Engineering and Sciences in Prince Sattam bin Abdulaziz University, Saudi Arabia, P.O. Box 151, Al-Kharj 11942, Saudi Arabia. E-mail: fayoumi66@yahoo.com*

cation Centre (HLR/AuC) in the home network can perform mutual authentication by sharing the same secret key in advance.

The UMTS AKA protocol is still vulnerable to security attack, such as redirection attack and man-in-the-middle (MITM) attacks. The redirection attack allows an adversary to redirect user traffic to an unwanted serving network and impersonating uncorrupted networks using authentication vectors from corrupted networks since the authentication vectors could be used by any serving network.. In this case, a false base station impersonates a serving network and then redirect user traffic from one SN to a SN of their choosing with lower security. UMTS AKA procedure will normally succeed and neither the home network nor the victim user can detect this attack.

Since the user will not notice being connected to another network and could cause billing problem to get unusually high bills as the service rates offered by different networks are not always the same. Additionally it might be used to redirect traffic to networks with lower security, causing an incorrect impression of the security level to be applied [13].

The MITM attack can occur when an adversary eavesdrops the data communication prior transmission at the session initiated by a valid MS. This attack occurs while a valid MS connecting to a GSM BSS. In this case, the adversary can intercept and inject some data. In UMTS, an adversary hides himself between the MS and Visit Location Register/Serving GPRS Support Node (VLR/SGSN) and tries to bypass the UMTS security and forces the victim MS to use the less secure GSM authentication and the obtains AUTN.

However, the UMTS AKA protocol has some shortcomings in the performance and complexity of operations. Firstly, the bandwidth consumption between visitor serving network and home network is inadequate when MS requests to authenticate itself for the VLR/SGSN in serving network (SN) and no authentication vectors are available the VLR/SGSN must turn back to HLR in the Home Network (HN) to make a registration request to generate another array of n authentication vectors. Moreover, when the subscriber roams to a newly visited VLR/SGSN within a different serving network domain the authentication vectors in the old VLR/SGSN are deleted, which is called an unused authentication vectors problem. Subsequently, as a lot of data being sent between VLR/SN and HLR/HN, this has impact on the performance of AKA protocol.

Secondly, as the generation of authentication vectors (AV) is expensive and which require the generation of five records in each AV will increase the delay time in home network. Thirdly, the storage space overhead occurs if there are m subscribers, and an array of n authentication vectors for each subscriber in SN, then the SN must wastefully store $n \times m$ authentication vectors. Fourthly, the management of sequence number (SEQ) which is needed for synchronization between mobile station and its home network, and the periodical authentication is achieved by comparing a SEQ counter value between an MS and a VLR/SGSN periodically. The SEQ is susceptible to synchronization failure. Therefore, both MS and HN are needed to maintain SEQ to accomplish that process between them [13].

To alleviate the weaknesses of the existing UMTS AKA protocol, this paper proposes a dynamic and secure authentication and key agreement protocol for UMTS. Therefore, the aims of the enhancement protocol are listed as follows.

1. To achieve bilateral authentication between MS and HLR in the home network.
2. To achieve bilateral authentication between MS and VLR in the serving network.
3. To reduce the stored space in VLR in the serving network.
4. To reduce the bandwidth consumption by reducing the authentication transmission overhead between VLR/SN and HLR/HN.
5. To provide stronger key agreement to protect the communication between the MS and the VLR/SGSN.
6. To defeat redirection and man-in-the-middle attacks and partially prevent DoS attacks

The remaining part of this paper is organized as follows. Section 2 , gives the literature review and related work. Section 3 describes detailed UMTS AKA protocol. The proposed protocol DS-AKA is presented in section 4. In Section 5, the security analysis for the proposed protocol is presented. The comparison with related work is presented in section 6. The paper is concluded in Section 7.

2 Related Work

Several authentication schemes have been proposed for mobile networks to enhance the security of mobile communication systems based on several authentication techniques. These techniques only provide some security features and have some weaknesses. Most of these schemes are based on the use of symmetric key cryptosystems and a challenge-response exchange. In this context, Many symmetric key based AKA protocols were proposed for UMTS network to improve the security of UMTS AKA and effective utilization of bandwidth during the authentication.

In 2005, Zhang and Fang [14] proposed a new authentication and key agreement protocol, which overcomes redirection attack and drastically lowers the impact of network corruption. The protocol is called adaptive protocol AKA (AP-AKA). They solve these problems by eliminating the need for synchronization between a mobile station and its home network and providing a way for the MS to verify if AV's are indeed coming from SN and have not been used before (i.e., The MS can verify for itself if an AV used by the VLR is a fresh one). This is achieved by sending additional identity information from the user and the serving network along with the authentication data. So if an attacker redirects messages of the MS the wrong ID_{SN} will be used and the ID_{SN} sent by the MS will not match the ID_{SN} sent by the VLR. However, AP-AKA protocol is still vulnerable to the middle-in-the- attack.

However, both UMTS-AKA and AP-AKA protocols have the problem of the bandwidth consumption between SN and HN. It is attractive to choose a suitable length (L) value for AV in the third generation mobile networks. So, many techniques are developed to minimize the authentication signalling cost and network bandwidth with consumption by selecting the dynamic length (L) for an authentication vector. Yet with this

improvement, Lin and Chen [15] and AL-Saraireh and Yousef [16] are still there are bandwidth consumption. Unfortunately, the performance drawbacks still strike as follows. First, the space overhead strikes when n AVs in the SN are being stored. Second, there is bandwidth consumption between SN and HN since HN needs to pass n AVs to SN. The two problems can be solved by several techniques.

Harn and Hsin [17] proposed an enhanced registration and AKA scheme for UMTS. By introducing a combination of hash-chaining and keyed HMAC techniques, in their proposed protocol they claim it can provide strong periodically mutual authentication, strong key agreement, and a non-repudiation service in a simple and elegant way. However, due to the underlying hash chaining technique [18], the security was enhanced while more computation overhead of hash chaining was incurred at MS and SN in each session. This could have a negative impact on the performance of this protocol. However, this protocol also does not clear the security issues against various attacks.

X-AKA protocol [19] was proposed an extension of the UMTS-AKA protocol to prune off the transmission of authentication vectors (AV) and improves its bandwidth utilization. In X-AKA, SN must continually generate random numbers to challenge MS to reply corresponding responses for every authentication. It is noticeable, random challenge generation overhead occurs in SN.

Both Harn-Hsin's and Huang-Li's protocols, the security for MS to identify the active SN is not mentioned so that an adversary can redirect the user traffic from the active SN to another SN. The redirection and man-in-the-middle attacks are not prevent in the two protocols.

Al-Saraireh and Yousef's protocol [20] primary emphasis on reducing the bandwidth for transmitted authentication vectors during authentication and therefore, the AVs are only generated by the MS instead of by the VLR. Al-Saraireh and Yousef's protocol eliminates the cost of delivering AVs . the protocol does not clear the security issues with redirection as well as man-in-the-middle attacks.

Ou, Hwang, and Jan [21] proposed a new protocol COCKTAIL-AKA, to overcome the congenital defects of UMTS AKA protocol. In this protocol, each service network produces its own AVs (MAVs) in advance. These MAVs are produced only once but can be reused later. While authenticating the MS, the HLR/AuC calculates a private authentication vector (PAV) for MS. The PAV is transferred to the SGSN. Then, the SGSN uses the PAV and MAV to generate several effective AVs for subsequent authentications. Cocktail-AKA is penetrable to DoS attack and impersonation attack [22]. It also does not solve the synchronization problem between MS and HLR.

Huang and et al [23] proposed a new protocol namely S-AKA, to defeat the redirection, man-in-the-middle and denial of service attacks. However, the S-AKA reduces bandwidth consumption up to 38% and also decreases the number of messages required in authenticating mobile subscribers. In S-AKA, SN must continually generate random numbers to challenge MS to reply corresponding responses for every authentication. It is noticeable, random challenge generation overhead occurs in SN. The NS-AKA protocol in [24] reduces the overheads, and is free from redirection and MITM attacks, but does not provide resistance against denial of service attack.

3 Description of UMTS AKA Protocol

The UMTS AKA protocol is performed in two procedures, as shown in figure 2 and 3 respectively. Firstly, Registration and Distribution Vectors which is called Authentication vectors from the home network (HN) to the serving network (SN): the MS registers with its HLR/AuC in the HN and then generate and distributes authentication vectors from the HLR/AuC to the VLR/SGSN in the SN. Secondly, Authentication and Key Agreement procedure runs between MS and VLR/SGSN. When the protocol is executed in the home network or when the serving network has unused authentication vectors for the user, the first phase is not executed.

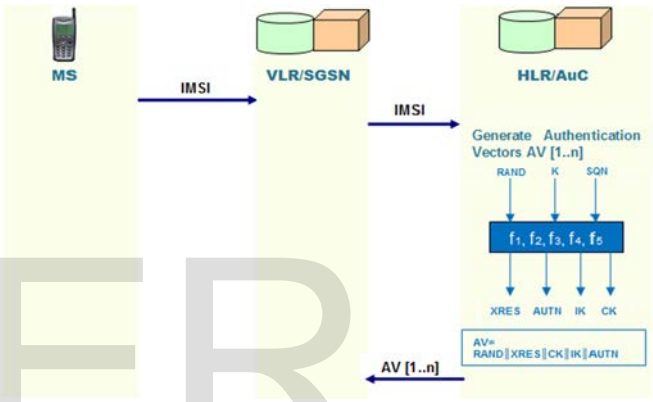

Fig. 2. UMTS AKA Phase1 – Distribution of Authentication Vectors

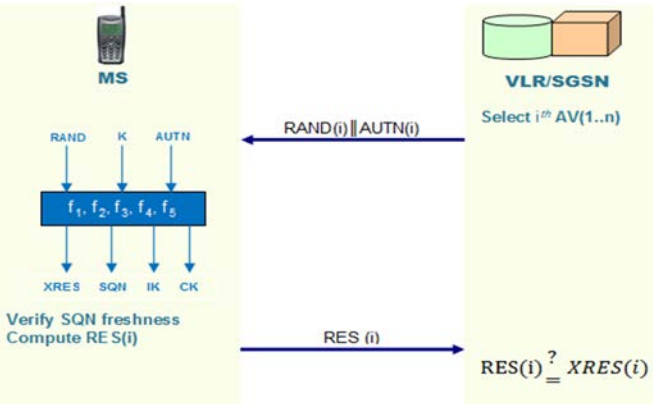

Fig. 3. UMTS AKA Phase1 – Distribution of Authentication Vectors

3.1 Distribution of Authentication Vectors

1. The MS sends a registration request to visitor location register/serving GPRS support node VLR/SGSN in the SN. The registration request includes an MS's IMSI.
2. Upon receiving the registration request, the VLR/SGSN in the SN passes the registration authentication request that is generated in step 1 to MS's HLR in the HN.
3. Upon receipt of a request from the VLR/SGSN, the HLR/AuC in the HN generates an ordered array of n

authentication vectors $AV(1 \cdots n)$ whose order is based on SQN_{HN}. It sends an authentication data response as an ordered array of n authentication vectors to the VLR/SGSN in the SN via a secure channel. The AV_S is generated by using the secret key K_i which is pre-shared with the subscriber. Each AV consists of $RAND, XRES$ (Expected Response), Ck (Cipher Key), IK (Integrity Key) and AUTN (Authentication Token).

4. Upon receipt of AV_S, the VLR/SGSN in the SN stores the authentication vectors for performing the subsequent authentication and key agreement procedure. Each AV is used for one authentication and key agreement between the VLR/SGSN in the SN and the MS.

3.2 Authentication and Key Agreement

1. In the i^{th} performance of the second procedure, VLR/SGSN in the SN initiates an authentication and key agreement to authenticate the MS, the VLR/SGSN selects the next authentication vector from the ordered array. It selects the i^{th} AV to run this procedure on a first-in/first-out (FIFO) basis. SN sends the parameters $RAND_i$ and $AUTH_i$ to the mobile station (MS). Therefore, one authentication vector is needed for each authentication instance. This means that the signaling between the VLR and AuC is not needed for every authentication event.
2. Upon receipt of $RAND_i$ and $AUTH_i$ where $(AUTN = SQN \oplus AK \| AMF \| MAC)$, the MS performs six steps.
   - Computes the anonymity key $AK = \int_K^5(RAND)$.
   - Retrieves the sequence number $SQN = (SQN \oplus AK) \oplus AK$.
   - Computes expected message authentication code $XMAC = \int_K^1(SQN, RAND, AMF)$ and compares this with MAC which is included in AUTN.
   - If they are different, the MS sends the user authentication rejection back to the VLR/SGSN with an indication of the cause and the user abandons the procedure
   - Otherwise, the MS confirms whether the freshness that the received sequence number SQN is in the correct range (i.e., $SQN_{HN} > SQN_{MS}$) or not. If the result is negative, the MS sends synchronization failure back to the VLR/SGSN including an appropriate parameter, and abandons the procedure.
   - Otherwise, the MS computes $RES = \int_K^2(RAND)$ and sends it back to VLR/SGSN in the SN. Meanwhile MS computes the cipher key $CK = \int_K^3(RAND)$ and the integrity key $IK = \int_K^4(RAND)$.
3. The VLR/SGSN makes verification by comparing the received RES with $XRES$. If they match, the mutual authentication and key agreement between the MS and the VLR/SGSN is completed successfully. The established keys CK and IK are stored in the VLR/SGSN and will then be transferred to the RNC when needed.
4. If a VLR/SGSN runs out of AV_S, it can request another ordered array of AV_S from the HLR/AuC.

4 Description of Proposed Protocol: Dynamic and Secure Authentication and Key Agreement Protocol (DS-AKA)

The proposed authentication protocol is divided into two procedures; the first one is called the initial authentication procedure, which flow from MS⇔VLR⇔HLR. The second one is limited between MS⇔VLR and is called the subsequent authentication procedure, as shown in Fingers 4 and 5, respectively.

4.1 First Procedure: DS-AKA-I

Step 1: M_1^I: $\{IMSI, Acc_m \oplus AK, RN_m, MAC_{m_i}\}$, MS sending a registration request to VLR/SN

When an MS needs to authenticate itself to all entities of network to access or utilize network services, the MS invokes the distribution of authentication procedure by sending the authentication request messages to the HLR/AuC through VLR in the serving network. In M_1^I, MS's IMSI represents the identity of a subscriber, LAI represents the identifier of the location area of the BSS, and it indicates the physical connection between the MS and BSS and Acc_m represents the number of successful MS authentication and is used to guarantee the freshness of authentication request. Acc_m , which is initially set to 0, increase on each successful authentication. Furthermore, in order to accomplish the authentication request message the MS will do the following processes:

1. Generate a random number RN_m.
2. Compute the temporary key $T_K = \int_K^x(RN_m \| LAI)$
3. AK represents the anonymity key to provide user identity confidentiality and computed as $AK = \int_{T_K}^5(RN_m)$.
4. MAC is the message authentication code for the MS and computed as: $MAC_{m_i} = \int_{T_K}^1(Acc_m \| LAI)$.

Step 2: M_2^I : $\{IMSI, Acc_m \oplus AK, RN_m, MAC_{m_i}, LAI\}$, SN passing the authentication Request Message to MS's HN.

When the VLR/SN receives the message from the MS, the VLR/SN is able to recognize the HLR/HN to which MS belongs by reading the IMSI; and then it passes the M_1^I to the intended HLR/HN together with BSS's LAI. The VLR/SN maintains a profile for that MS under the identity of user (IMSI) which contains the privileges of a registered user for subsequent authentication. So, The VLR/SN waits to receive the authentication result from HLR/HN.

Step 3: M_3^I : $\{AUTN_H, T_K, n\}$, Verifying the MS and sending authentication token and temporary key to SN.

Upon receipt of the M_2^I, the HLR/AuC in the home network verify the MS according to the information that have been received, and then builds the Authentication Data Response message for MS and VLR/SN. In order to accomplish the authentication process, HLR/AuC will do the following:

1. Compute the temporary key $T_K = \int_K^x(RN_m \| LAI)$
2. Computes the anonymity key $AK = \int_{T_K}^5(RN_m)$.
3. Retrieves the $Acc_m = (Acc_m \oplus AK) \oplus AK$
4. Computes expected message authentication code $XMAC_{m_i} = \int_{T_K}^1(Acc_m \| LAI)$ and compares this with MAC_{m_i} which is included in M_2^I. By checking MAC_{m_i}, the HLR/AuC can verify whether the *LAI* reported by the VLR/SGSN is the same as that recognized by the MS. If they are different, the HLR/AuC rejects the request with an indication of the cause and the user abandons the procedure.
5. Otherwise, the HLR/AuC confirms whether the fresh-

ness that the received Acc_m is in the correct range (i.e., $Acc_H > Acc_m$) or not. If the result is positive, the HLR/AuC considers it a replay and abandons the procedure.

6. Otherwise, HLR/AuC generates a random number RN_H and derive a message authentication code by hashing RN_H and AMF (authentication management field) as: $MAC_H = \int_K^1(RN_H \| AMF)$. After that, the HLR/HN concatenates the aforementioned token to derive $AUTN_H = (MAC_H \| RN_H \| AMF)$ and send it to VLR/SN together with T_K and n via secure channel, where n is the life time of the T_K.

Step 4: M_4^I: $\{AUTN_S\}$, generating challenge information for MS
When the VLR/SN receives the response message from HLR/HN, it means that the MS has proved itself to its HN successfully. Therefore, the VLR/SN stores the authentication vector for performing the subsequent authentication. In order for the MS to verify the authenticity of the SN in the subsequent authentication AKA procedure, the SN generates response information for the MS and sends them to MS. The VLR/SGSN performs the following steps:

1. The VLR/SGSN increments its Acc_S by 1 and generates a random number RN_S.
2. Computes the anonymity key $AK = \int_{T_K}^5(RN_S)$.
3. Compute reclusively a hash chaining authenticators $MAC_{S_i} = \int_{TK}^1(MAC_{S_{i-1}} \| Acc_S)$, where $MAC_{S_{i-1}} = \int_{TK}^1(MAC_H \| RN_S \| RN_H \| n)$.
4. Construct $AUTN_S = (MAC_{S_i} \| RN_S \| RN_H \| AMF \| Acc_S \oplus AK \| n)$ and sent it to MS.

Step 5: M_5^I: $\{XRES\}$, SN authenticating the MS.
Upon receipt of $AUTN_S$ from VLR/SGSN, the MS authenticates VLR/SGSN, HLR/AuC by deriving and verifying MAC_S and MAC_H. The deriving and verifying steps are as follows:

1. Retrieves the $Acc_S = (Acc_S \oplus AK) \oplus AK$
2. Compute expected message authentication code $XMAC_H = \int_K^1(RN_H \| AMF)$, where RN_H and AMF are retrieve from $AUTN_S$ in step 4.
3. Compute the hash chaining authenticators $XMAC_{S_i} = \int_{TK}^1(MAC_{S_{i-1}} \| Acc_S)$, where $MAC_{S_{i-1}} = \int_{TK}^1(MAC_H \| RN_S \| RN_H \| n)$.
4. MS sets the new Acc_s to its Acc_m.
5. Compute an expected response message as: $XRES = \int_{T_K}^2(MAC_{S_{i-1}} \| Acc_S)$ for mutual authentication and sent $XRES$ to VLR/SGSN. Meanwhile the MS Computes $IK_i = \int_{T_K}^3(IK_{i-1} \| Acc_S)$ and $CK_i = \int_{T_K}^4(CK_{i-1} \| Acc_S)$ where $IK_0 = \int_{T_K}^3(RN_S \| Acc_S)$ and $CK_0 = \int_{T_K}^4(RN_S \| Acc_S)$ respectively.
6. Upon receipt of $XRES$, VLR/SGSN authenticates the MS by verifying the $XRES ? = RES = \int_{T_K}^2(MAC_{i-1} \| Acc_m)$. If the MS is successfully authenticated, then the VLR/SGSN computes IK_i and CK_i to protect the communication between the MS and the VLR/SGSN subsequently.
7. As aforementioned, the man-in-the-middle attack occurs on UMTS network due to its interoperability with GSM network. The security weaknesses of GSM expose the entire mobile system to this attack. To prevent the man-in-the-middle attack, the DS -AKA protocol employs an extra key DS_K when the GSM BSS involved in a conversation. Both the MS and the VLR/SGSN compute $DS_{K_i} = \int_{T_K}^6(DS_{K_{i-1}} \| Acc_S)$, where $DS_{K_0} = \int_{T_K}^6(RN_S \| Acc_S)$ and can use DS_K key to protect the confidentiality of the data passing through the GSM BSS and prevents the communication from being eavesdropped.

4.2 Second Procedure: DS-AKA-II

After the initial authentication, both MS and VLR/SN possess authentication information and the temporary key T_K that it share with each other and subsequently can accomplish the mutual authentication by itself without intervention of HLR/AuC. That is, subsequent authentication only happens between the MS and the SGSN/VLR using three message exchanges. In the i^{th} performing the second procedure, the authentication is described as follows:

Step 6: M_1^{II}: $\{TMSI, MAC_{m_i}, Acc_m \oplus AK\}$
MS increments its Acc_m by 1

1. Compute reclusively a hash chaining authenticators $MAC_{m_i} = \int_{TK}^1(MAC_{m_{i-1}} \| Acc_m)$, where $MAC_{m_{i-1}}$ has been computed in step 1 of the first procedure.
2. Construct M_1^{II}: $\{TMSI, MAC_{m_i}, Acc_m \oplus AK\}$ and send it to VLR/SGSN.

Step 7: M_2^{II}: $\{AUTN_S\}$

1. VLR/SGSN retrieves the $Acc_m = (Acc_m \oplus AK) \oplus AK$ and increments its Acc_S by 1 and compares it with the Acc_m to check if it is a replay.
2. On behalf of the HLR/AuC, the VLR/SGSN verify whether the equation the recursive hash chaining authenticators $XMAC_{m_i} = \int_{TK}^1(MAC_{m_{i-1}} \| Acc_m)$ is hold, VLR/SGSN updates MAC_{m_i} and store it for the next visit to generate the $MAC_{m_{i+1}}$.
3. If the $XMAC_{m_i}$ is legitimate, the VLR/SGSN computes a new sequence of hash chaining authenticator $MAC_{S_i} = \int_{T_K}^1(MAC_{S_{i-1}} \| Acc_{S_i})$, where $i \leq n$ and $MAC_{S_{i-1}}$ computed in the previous visit.
4. Construct $AUTN_S = (MAC_{S_i} \| Acc_S \oplus AK)$ and sent it to MS.

Step 8: M_3^{II}: $\{XRES\}$

1. MS retrieves the $Acc_S = (Acc_S \oplus AK) \oplus AK$
2. MS authenticate VLR/SGSN by verifying $XMAC_{S_i} = \int_{T_K}^1(MAC_{S_{i-1}} \| Acc_{S_i})$.
3. MS computes $XRES = \int_{T_K}^2(MAC_{S_{i-1}} \| Acc_S)$ and send it to VLR/SGSN.
4. On receiving, the VLR/SGSN authenticates the MS by verifying the freshness and correctness of $XRES$. Now, both the MS and VLR/SGSN can use IK, CK and DS_K keys for the purpose of confidentiality, integrity and encryption.

IJSER

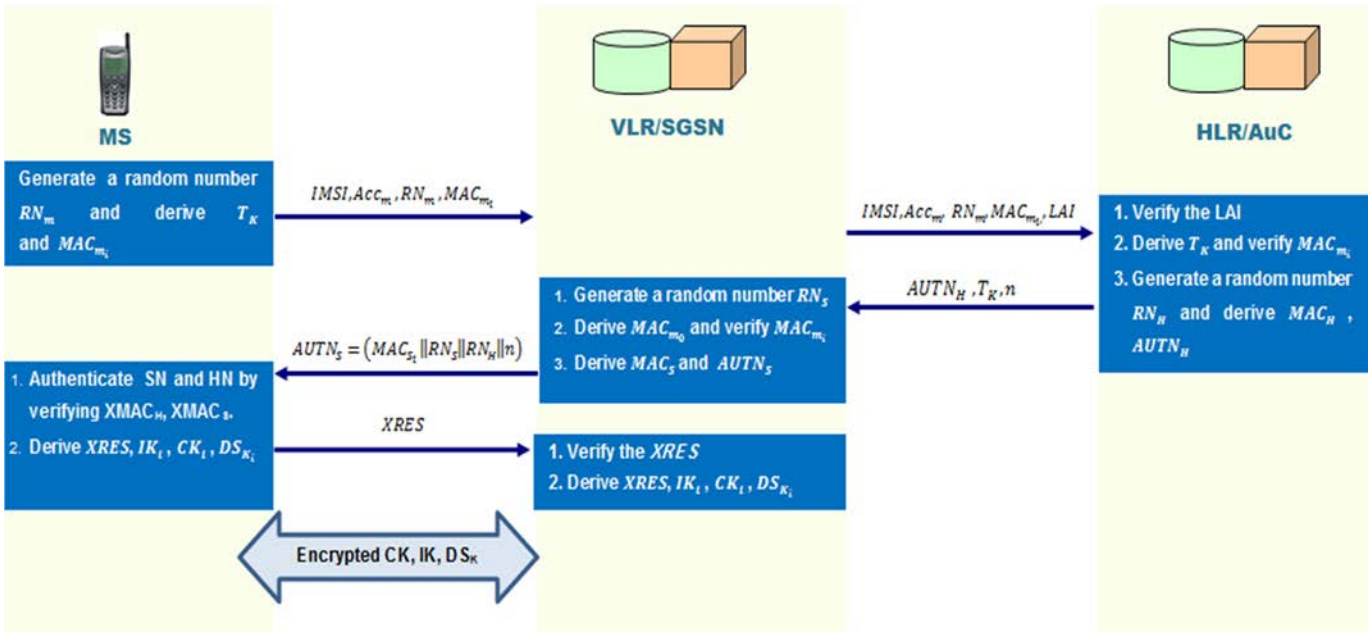

Fig. 4. DS-AKA Phase I – Initial Authentication Procedure

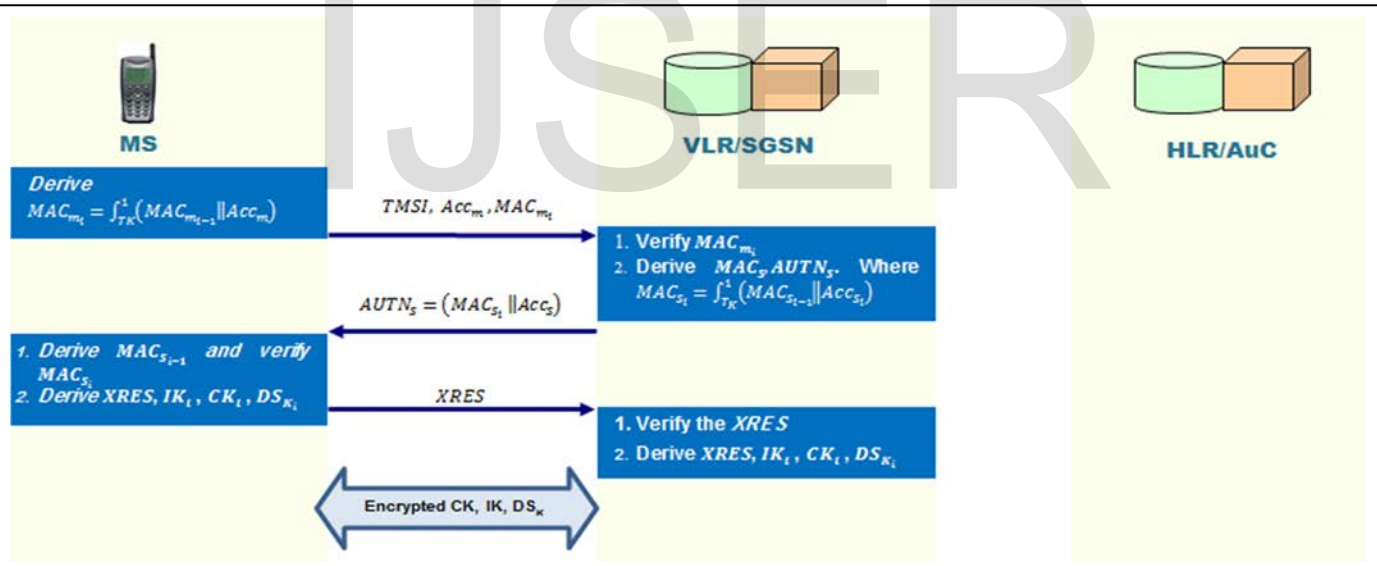

Fig. 5. DS-AKA Phase II – Subsequent authentication procedure

5 Security Analysis for Proposed Protocol

The proposed DS-AKA protocol adopts the framework of UMTS AKA, and thus it also inherits the security issues of UMTS AKA such as entity authentication between an MS and an VLR/SGN, data integrity and user traffic confidentiality. Furthermore, the transmitting channel between VLR/SGSN and HLR/Auc is assumed to be secure. The adopting the same architecture of UMTS helps DS-AKA resist various network attacks.

The proposed DS-AKA protocol has been developed with the aim of keeping the complexity of this function as low as possible, and providing a high level of security and efficiency for the bandwidth used. A detailed analysis has been made of how the proposed scheme meets the security requirements and how examine additional security issues in the proposed DS-AKA protocol. Furthermore, explain how DS-AKA deters intruders from redirection, man-in-the-middle and DoS attacks.

5.1 Mutual Authentication

- Mutual authentication between MS and its HN: The UMTS AKA provides procedure for mutual authentication of the MS and serving system. It is clear in the UMTS AKA protocol, HLR/AuC in HN has no mechanism to authenticate MS. In contrast, the DS-AKA protocol, HLR/AuC authenticate MS. In step 3 of the DS-AKA protocol, the HLR/AuC authenticates and confirm the identity of the MS by verifying Acc_m and the Ms' message authentication code MAC_{m_i} on receipt M_2^I. Note that the Acc_m is used to guarantee the freshness of the authentication request and MAC_{m_i} is used to protect the integrity of Acc_m and LAI. Therefore, the verifying the MAC_{m_i} can resist the replay and redirection attacks.

In step 5 of the DS-AKA protocol, the MS authenticates VLR/SGSN, HLR/AuC by deriving and verifying MAC_S and MAC_H. The MS checks $AUTN_s$ that received in M_4^I which is contained MAC_{S_i}, RN_S, RN_H, AMF, Acc_S , n. The MS can compute expected authentication code of HLR/AuC as $XMAC_H = \int_K^1(RN_H \| AMF)$, and then compute expected authentication code of VLR/SGSN as $XMAC_{S_i} = \int_{TK}^1(MAC_{S_{i-1}} \| Acc_S)$, where $MAC_{S_{i-1}} = \int_{TK}^1(MAC_H \| RN_S \| RN_H \| n)$. If the $XMAC_{S_i}$ is equal to MAC_{S_i}, it means that VLR/SGSN and HLR/AuC are authenticated successfully.

Therefore, the MS confirms the authenticity of the VLR/SGSN and HLR/AuC together. After the initial authentication during the origination and termination call, the VLR/SGSN gets a secret temporary key T_K that it shares with the MS and subsequently can accomplish the mutual authentication by itself even when the HLR/AuC is not involved, the MS can recursively calculate the hash chaining authenticator $MAC_{S_{i-1}} = \int_{TK}^1(MAC_H \| RN_S \| RN_H \| n)$. Therefore, the MS is still authenticate the HLR/AuC with M_2^{II} in the subsequent authentication procedure.

- Mutual authentication between MS and its SN: Similar to authenticating the HLR/AuC, on receiving $AUTN_s$ that received in M_4^I, the MS computes expected authentication code of HLR/AuC $XMAC_H$ and the expected hash chaining authenticator $XMAC_{S_i}$ and compares it with the received MAC_H, $XMAC_{S_i}$. If both are equal then the authentication of SN is successful. In step 5 of the DS-AKA protocol upon receipt of M_5^I, VLR/SGSN authenticates the MS by verifying the $XRES \stackrel{?}{=} RES$. If the equation is hold then MS is successfully authenticated. This ensures mutual authentication between MS and VLR/SGSN.
- Freshness of session keys: The DS-AKA protocol provide stronger key agreement to protect the communication between the MS and the VLR/SGSN. In the DS-AKA-I procedure, the CK_i and IK_i are generated in step 4 and step 5, whereas in DS-AKA-II procedure they are generated in step 7 and step 8. Since the composition of session keys CK and IK are based on the previous hash changing output and the accumulated Acc_S, the freshness of these keys can be guaranteed by CK_{i-1} . and IK_{i-1} . The CK_{i-1} and IK_{i-1} can be taken as the new random challenge for the input of the current hash chaining CK_i and IK_i. The Acc_S in M_4^I or M_2^{II} is accumulated on each successful authentication and the anonymity key AK is used to conceal these parameters. Therefore, the Acc_S can be used to guarantee the freshness of these keys.

5.2 Temporary Key

The DS-AKA protocol employs a key authentication as a temporary key, which is generated during the execution of the registration process and caches in the MS and VLR/SGSN in the serving network for the subsequent authentication process (intra-network). The new mechanism significantly reduces the communication overhead between home and serving networks for roaming authentication.

The DS-AKA protocol allows the VLR/SGSN to perform most of the authentication and agreement procedure. The temporary key is generated during registration and both the HN and MS can compute this key. The temporary key can then be used by the SN as A-Key, effectively reducing the need for subsequent communications with the HN. In addition the SN will not have to store a number of AV's, because it can calculate them when needed. Therefore, the proposed protocol uses another key generation function f^x, which generates a 128-bit or higher hash result, to get a better security level.

The temporary key T_K is generated as $T_K = \int_K^x(RN_m \| LAI)$. The parameter K is the secret shared key by MS and its HN. Although the RN_m is transmitted as cleartext and the key generation function f^x is a public cryptography generator and the T_K is assumed to be securely sent from HLR/AuC to VLR/SGSN, the temporary key cannot be generated without knowing the secret key K shared by MS and HLR/AuC. After the initial authentication, the VLR/SN gets a temporary key T_K that it shares with the MS and subsequently can accomplish the mutual authentication by itself. Therefore, VLR/SGSN owns the temporary key to authenticate MS on behalf of the HLR/AuC.

In the DS-AKA protocol, the temporary key is used for calculating a recursive hash chaining authenticators $MAC_{S_i} = \int_{TK}^1(MAC_{S_{i-1}} \| Acc_S)$, $MAC_{m_i} = \int_{TK}^1(MAC_{m_{i-1}} \| Acc_m)$ and $XRES = \int_{T_K}^2(MAC_{S_{i-1}} \| Acc_S)$. The Acc_m and Acc_S are accumulated on each successful authentication and the anonymity key AK is used to conceal these parameters, the Acc_m and Acc_S can be used to guarantee the freshness of authentication request.

Since MAC_{S_i} , MAC_{m_i} and $XRES$ are derived from the previous visit, , the freshness of hash chaining authenticators can be guaranteed by Acc_m and Acc_S. Therefore, MAC_{S_i} , MAC_{m_i} and $XRES$ are changed in each authentication. If an intruder eavesdrops these authenticators to generate the expected hash chaining authenticators such as $MAC_{S_{i+i}} = \int_{T_K}^1(MAC_{S_i} \| Acc_{S_i})$, $MAC_{m_{i+1}} = \int_{TK}^1(MAC_{m_i} \| Acc_m)$, $XRES = \int_{T_K}^2(MAC_{S_i} \| Acc_S)$, its infeasible without knowing the temporary key T_K. Moreover, if an intruder eavesdrops these authenticators to reverses them to derive the temporary key, it is infeasible because all cryptography function that used are a one way function.

5.3 Resistance to Attacks

- **Redirection Attack:** In UMTS-AKA, an authentication vector (AV) can be used by any SN. This situation can be abused to redirect data to an SN. This is called a redirection attack. In this case a false base station impersonates a SN. The false base station will then redirect all traffic to a SN of their choosing. The AKA procedure will normally succeed and the user will not notice being connected to another network. This

can cause the user to get unusually high bills. Additionally it might be used to redirect traffic to networks with lower security, causing a wrong impression of the security level applied.

In DS-AKA protocol, an authentication vector (AV) generated by the user's HN can only be used by a particular serving network (SN). This is achieved by involving the identity of location area LAI of the BSS for SN in the generation and verification of the message authentication code MAC_m. Whenever an MS enters a new SN will get the identity of that SN, and then start a registration process to register itself to the network by replying with a Acc_m and RN_m together with $MAC_{m_i} = \int_{T_K}^{1}(Acc_m \| LAI)$ providing integrity of Acc_m , LAI in M_1^I and store a profile of the new SN in its database, which includes Acc_m and RN_m.

Since the SN knows the LAI of the BSS forwarding M_1^I, then forward that message together with BSS's LAI to HN. When the HN receives the user's authentication request from the SN, the HN verifies the MAC_{m_i} to ensure that the user is indeed in the territory of the SN by matching the LAI that reported in M_2^I from VLR/SGSN. When the HN begins to generate the authentication vector, it should insert T_K, n and $AUTH_H = (MAC_H \| RN_H \| AMF)$ into the authentication vector AV, where T_K is a temporary key and n is the life time of the T_K. Furthermore, the LAI embedded in $T_K = \int_K^x (RN_m \| LAI)$ which is computed both in MS and HN.

Therefore, the DS-AKA protocol prevent the redirection attack by preventing a user from being tricked and redirect traffic to a network with lower security and solve the mischarged billing problem.

- **Man-in-the middle Attack**: In DS-AKA, a new key DS_K is used to encrypt and decrypt the communicated information between the MS and its BTS. Since the original UMTS AKA does adopted a key generation for such key, a new key generation function f^6 is used to generate DS_K key between the MS and VLR/SGSN to prohibit the communication from being eavesdropped or altered. The new key DS_K is generated after exchanging the messages M_4^I , M_5^I in initial authentication procedure and M_2^{II}, M_3^{II} in subsequent authentication procedure. Therefore, the DS-AKA prevents the man-in-the-middle attack by encrypting the information communication before the transmission and then the data confidentiality of the communication channel between the MS and the VLR/SGSN can be guaranteed.
- **DoS Attack:** When an MS needs to authenticate itself to all entities of network to access or utilize network services, the MS invokes the distribution of authentication procedure by sending the authentication request messages to the HLR/AuC through VLR/SGSN in the serving network. Firstly, The authentication between the MS and his HLR/AuC relies on the use of its message authentication code MAC_{m_i} which involved in M_1^I. This message provides the legal proof of the MS's intent to register itself. In SD-AKA-I procedure, the HLR/AuC could check the integrity of code MAC_{m_i}, so the forged message generated by a malicious using M_1^I can be detected by HLR/AuC on receipt of M_2^I from VLR/SGSN. Therefore, the DoS attack will be recognized at step 3 by HLR/AuC side and no more traffic would be procreated in the network.

Secondly, after the initial authentication, the VLR/SN gets a temporary key T_K and other authentication information that it shares with the MS and subsequently can accomplish the mutual authentication by itself without intervention of the HLR/AuC. That is, subsequent authentication (SD-AKA-II) only happens between the MS and the SGSN/VLR using three message exchanges M_1^{II}, M_2^{II}, and M_3^{II}. Consequently as explained in SD-AKA-II procedure at step7, the VLR/SGSN checks the integrity of the MAC_{m_i} in the message M_1^{II} with the temporary key T_K authorized by the HLR/AuC. In this case, if a malicious forges M_1^{II} and then send it to VLR/SGSN, the forged message can be immediately detected by the VLR/SGSN and then rejects the forged messages.

In the DS-AKA protocol, the temporary key is used for calculating a recursive hash chaining authenticators $MAC_{S_i} = \int_{TK}^{1}(MAC_{S_{i-1}} \| Acc_S)$, $MAC_{m_i} = \int_{TK}^{1}(MAC_{m_{i-1}} \| Acc_m)$ and $XRES = \int_{T_K}^{1}(MAC_{S_{i-1}} \| Acc_S)$. The Acc_m and Acc_S are accumulated on each successful authentication and the anonymity key AK is used to conceal these parameters, the Acc_m and Acc_S can be used to guarantee the freshness of authentication request. In the i^{th} performing the DS-AKA protocol, the VLR/SGSN accumulate the Acc_S by 1 and send it to the MS together with the MAC_{S_i}. The MS retrieves the Acc_S and checks if it is a replay and sets the new the Acc_S to Acc_m for synchronizing the Acc_m and Acc_S. the MS will set the new the Acc_S to Acc_m. Moreover, the management of synchronized Acc_S to Acc_m is needed for synchronization between an MS and a VLR/SGSN periodically and could them decide about the genuineness the messages by verifying the recursive hash chaining authenticators integrity. Meanwhile, the synchronization procedure could be used to detect the forged message of the M_1^{II} initiated by a malicious entity at VLR/SGSN side in DS-AKA-II.

Therefore, the DS-AKA can partially prevent DoS attacks . The forged message of the M_1^{II} can be immediately detected by the VLR/SGSN side in the subsequent authentication procedure DS-AKA-II, and then rejects the forged messages. While as, the forged message of M_1^I can only be detected by the HLR/AuC side in the initial authentication procedure DS-AKA-I.

6 Comparison With Related Work

As described in section 3, the UMTS AKA protocol consists of two procedures. First, the user registers with its HN, and then distributes an authentication vector (AVs) from the HLR/AuC in the home network to the VLR/SGSN in the serving network (SN). An AV is temporary authentication data which enables a VLR/SN to engage in UMTS authentication and key agreement with a particular user. Second, the authentication and key agreement procedure runs between MS and SN. The SN uses the authentication information of the first procedure to carry out the bilateral authentication between MS and SN and then an agreed key and a cipher key are provided. In UMTS, Obviously, the UMTS AKA protocol has the problems of the bandwidth consumption between SN and HN and the storage space overhead for SN's database.

Firstly, the bandwidth consumption between SN and HN occurs when MS requests to authenticate itself for the VLR/SGSN in serving network (SN) and no authentication

vectors are available the VLR/SGSN must turn back to HLR in the Home Network (HN) to make a registration request to generate another array of n authentication vectors. Moreover, when the subscriber roams to a newly visited VLR/SGSN within a different serving network domain the authentication vectors in the old VLR/SGSN are deleted, which is called an unused authentication vectors problem. Subsequently, as a lot of data being sent between VLR/SN and HLR/HN, this has impact on the performance of AKA protocol.

Since the UMTS AKA, AP-AKA and Harn&Hsia protocols are restricted to n times, the n authentication vectors are necessarily transferred from HN to SN for authentication. Obviously, these protocols have the problem of the bandwidth consumption between SN and HN. Whereas, the proposed DS-AKA, X-AKA and S-AKA protocol employ the temporary key technique to solve the reduction of bandwidth consumption problem by sending SN a temporary key, SN can directly authenticate MS without intervention of HN. Therefore, it is unnecessary for the VLR/SN to request another set of authentication vectors from HN and then reduce the network traffic.

Secondly, the storage space overhead occurs if there are m subscribers, and an array of n authentication vectors for each subscriber in SN, then the SN must wastefully store $n \times m$ authentication vectors. In UMTS AKA, AP-AKA protocols do not consider storage space overhead for SN's database, where an array of n authentication vectors for each MS must be stored in the VLR/SN. So if there are m MSs in a VLR/SN, then the VLR/SN must store $n \times m$ authentication vectors. Therefore, a space overhead occurs. In contrast, the proposed DS-AKA, X-AKA and S-AKA protocols employ the temporary key technique which make VLR/SN store only one copy of authentication vectors to authenticate MS instead of n copies of authentication vectors. Moreover, in Harn&Hsia only stores the authenticator $h^{n-i}(m)$ to authenticate MS. Therefore, the DS-AKA protocol reduces the storage space overhead for SN's database and then attains the reduction of VLR storage.

Thirdly, the design of the X-AKA and S-AKA protocol have a shortcoming in the complexity of operations in terms of random number generation overhead. In the i^{th} performing of the second procedure for both protocols, SN must continually generate a random number RN_s and computes the corresponding message authentication code MAC_s as a challenge information for MS as well as computes cipher key CK and integrity key IK. Therefore, the generation random number for each authentication has a negative impact on the performance of AKA protocol by increasing the computation and storage cost on SN side. Whereas the proposed DS-AKA protocol employs the key hash chaining technique. This technique does not need to continuously generate a new random challenge for each authentication iteration. In the i^{th} performing of the second procedure, SN produces a new sequence of hash chaining authenticator MAC_{S_i} by computing $MAC_{S_i} = \int_{T_K}^{1}(MAC_{S_{i-1}} \| Acc_{S_i})$, where $i \leq n$ and $MAC_{S_{i-1}}$ computed in the previous visit. Therefore, The $MAC_{S_{i-1}}$ can be used as a new random challenge for the input of a new sequence for keyed hash chaining. So, performing the AKA procedure between MS and SN for i^{th} times without intervention of the HLR in the home network and continuously generation a new random challenge.

As Table 1 shows, several authentication schemes had been proposed. All schemes have the properties, including mutual authentication between MS and HE, mutual authentication between MS and SN, reduction of bandwidth consumption between SN and HN, reduction of storage space for SN's database, Random Number Generation, Redirection Attack, Man-in-the middle Attack and DoS Attack.

TABLE 1
A Comparison among the AKA Protocols

Comparison Item	UMTS AKA	AP-AKA	Harn-Hsin	X-AKA	S-AKA	DS-AKA
MA_MS-HN	No	No	No	Yes	Yes	Yes
MA_MS-SN	Yes	Yes	Yes	Yes	Yes	Yes
RBC	No	No	No	Yes	Yes	Yes
RSSO	No	No	Yes	Yes	Yes	Yes
NS	Yes	No	No	No	No	No
RNG	$O(1)$	$O(1)$	$O(1)$	$O(n)$	$O(n)$	$O(1)$
RA	No	Yes	NO	No	Yes	Yes
MIMA	No	No	No	No	Yes	Yes
ReA	Yes	Yes	Yes	Yes	Yes	Yes
DoS	No	No	No	P	P	P

MA_MS-HN: Mutual Authentication between MS and HN; MA_MS-SN: Mutual Authentication between MS and SN: RBC: Reduction of bandwidth consumption between SN and HN; RSSO: Reduction of storage load for SN's database; NS: Need Synchronization between MS and HN; RNG: Random Number Generation Load , RA: Redirection Attack, Y: Robust; N:Not robust; P: Partially, MIMA: Man-in-the middle Attack, ReA: Replay Attack, DOS: DoS Attack

7 Conclusion

In this paper, the security vulnerabilities of UMTS AKA protocol are analysed and proposed a new dynamic and secure Authentication and Key Agreement Protocol by using the key hash chaining technique and the temporary key mechanism, called DS-AKA for UMTS networks.

Comparing with other protocols, the results shows that the proposed DS-AKA protocol improve the performance AKA protocol by reducing the communication times and fulfil the security requirements of the third generation mobile systems. The proposed protocol significantly reduces the communication overhead between the home network and the visited network, reduces the storage space overhead for SN's database and reduces the random number generation overhead. In addition, the proposed DS-AKA protocol can not only improve the performance of UMTS AKA but also withstand for both

redirection and man-in-the-middle attacks and can partially prevent DoS attacks. Future work will provide a performance analysis of DS-AKA protocol.

REFERENCES

[1] K. Passerini,, K., Bartolacci, R., and J. Fjermestad, "Reflections and Trends in the Expansion of Cellular Wireless Services in the U.S. and China," *Proc. communications of the ACM*. 50(10), 2007.

[2] S. Yuan, B. Elizabeth, Gao, X., and K. James, "Real-time traffic support in heterogeneous mobile networks," *Proc. Springer Science + Business Media, Wireless Network13*, pp.431–445, 2007.

[3] M. Iftikhar, B. Landfeldt,, and M. Caglar, "Traffic Engineering and QoS Control between Wireless DiffServ Domains Using PQ and LLQ," *Proc. ACM 978-1-59593-809-1, MobiWac'07, Chania, Crete Island, Greece*, 2007.

[4] 3GPP, "3G Security; Security Threats and Requirements: 3rd Generation Partnership Project (3GPP)," Technical Specification Group Services and System Aspects, 3GPP TS 21.133, 4.1.0 (2001-12) (Release 4), 2001.

[5] W. Stalling, *Cryptography and Network Security*: *Principles and Practice*. 6th edition. USA: Prentice Hall, 2014.

[6] M. Al-Fayoumi, N. Shilbayeh, "Cloning SIM Cards Usability Reduction in Mobile Networks," *Springer's Journal of the Network and Systems Management*, Vol. 22, no. 2, pp. 259-279, April 2014, doi: 10.1007/s10922-013-9299-8.

[7] ETSI,. "Recommendation GSM 03.20: Security related network functions", Technical report. European Telecommunications Standards Institute, ETSI, 1993.

[8] M. Rahnema., "Overview of the GSM System and Protocol Architecture," *IEEE Communication Magazine*, Vo. 31, no. 4, pp.92-100, 1993.

[9] L. Harn, and H.Y. Lin, "Modification to enhance the security of the GSM protocol," *Proc. 5th National Conference on Information security*, Taipei, Taiwan, pp. 41-20, 1995.

[10] K. Al-Tawil, A. krami, and H. Yousef, "A New Authentication Protocol for GSM Networks," *Proc. 23rd Annual IEEE Conference on Local Computer Networks (LCN'98)*, pp. 21-30, 1998.

[11] C.C. Lo, and Y.J. Chen, "Secure communication mechanisms for GSM networks," *IEEE Transactions on Consumer Electronics*, Vol. 45, no. 4, pp.1074-1080, 1999.

[12] C. Lee, M. Hwang, and W. Yang, "Extension of Authentication Protocol for GSM.," *Proc. IEE Communication*, Vol. 150, no. 2, pp.91-95, 2003.

[13] M. Al-Fayoumi, M. Alnababteh, M. Daoud, M. Alhawarat, "Dynamic Authentication Protocol for Mobile Networks Using Public-Key Cryptography," *International Journal of Science and Research (IJSR)*, Vol. 4, no. 1, January 2015.

[14] M. Zhang, Y. Fang, "Security analysis and enhancements of 3GPP authentication and key agreement protocol," *IEEE Transactions on Wireless Communication*, Vol. 4, no. 2, pp. 734–742, 2005.

[15] Y. Lin, Y. Chen, "Reducing Authentication Signaling Traffic in Third-Generation Mobile Network," *IEEE Transactions on Wireless Communications*, Vol. 2, no. 3, pp. 493-501, 2003.

[16] J. Al-Saraireh, S. Yousef, "Analytical Model: Authentication Transmission Overhead Between Entities in Mobile Networks," *Elsevier, Computer Communications Journal*, Vol. 30, no. 9, pp. 1713-1720, 2007.

[17] L. Harn, W. Hsin, "On the Security of Wireless Network Access with Enhancements," *Proc. 2003 ACM workshop on Wireless Security*, San Diego, USA, pp. 88-95, 2003.

[18] L. Lamport, "Password authentication with insecure communication," *Communication of ACM*, Vol. 24, no. 11, pp. 770-772, 1981.

[19] C. Huang C., J. Li, "Authentication and Key Agreement Protocol for UMTS with Low Bandwidth Consumption," *Proc. 19th International Conference on Advanced Information Networking and Applications (AINA'05)*, pp. 392-397, 2005.

[20] J. Al-Saraireh, S. Yousef, "A New Authentication Protocol for UMTS Mobile Networks," *EURASIP Journal on Wireless Communications and Networking*, Vol. 2006, no. 2, pp. 19-30, 2006.

[21] H. H. Ou, M. S. Hwang, J. K. Jan, "A cocktail protocol with the authentication and key agreement on the UMTS," *Journal of Systems and Software*, Vol. 83, no. 2, pp. 316–325, 2010.

[22] S. Wu, Y. Zhu, Q. Pu, "Security analysis of a cocktail protocol with the authentication and key agreement on the UMTS," *Communication Letters*, Vo. 14, no. 4, pp. 366–368, 2010.

[23] Y. L. Huang, C. Y. Shen, S. W. Shieh, "S-AKA: A provable and secure authentication key agreement protocol for UMTS networks," *IEEE Transactions on Vehicular Technology*, Vo. 60, no. 9, pp. 4509–4519, 2011.

[24] N. Saxena, N. S. Chaudhari, "NS-AKA: An improved and efficient AKA protocol for 3G (UMTS) networks," *Proc. International conference on advances in computer science and electronics engineering (CSEE'14)*, Kuala Lampur, Malaysia, pp. 220–224, 2014.

IJSER