Private Identification, Authentication, and Key Agreement Protocol with Security Mode Setup

Farshid Farhat, Somayeh Salimi, Ahmad Salahi

***Abstract*— Identification, authentication and key agreement protocol of UMTS networks with security mode setup has some weaknesses in the case of mutual freshness of key agreement, DoS-attack resistance, and efficient bandwidth consumption. In this article we consider UMTS AKA and some other proposed schemes. Then we explain the known weaknesses of the previous frameworks suggested for the UMTS AKA protocol. After that we propose a new protocol called private identification, authentication, and key agreement protocol (PIAKAP), for UMTS mobile network. Our suggested protocol combines identification and AKA stages of UMTS AKA protocol while eliminates disadvantages of related works and brings some new features to improve the UMTS AKA mechanism. These features consist of reducing the interactive rounds of the UMTS AKA with security mode setup and user privacy establishment.**

***Index Terms*—Identification, Authentication, Key Agreement, Privacy, UMTS, Mobile Network, Security Protocol**

I. ABBREVIATIONS

3GPP	Third-Generation Partnership Project
AK	Anonymity Key
AKA	Authentication and Key Agreement
AMF	Authentication Management Field
AREP	Authentication Reply
AREQ	Authentication Request
AuC	Authentication Center
AUTN	Authentication Token
AV	Authentication Vector
BSC	Base Station Controller
BTS	Base Transceiver Station
CK	Cipher Key
CM	Connection Management
CS	Circuit Switched
DoS	Denial of Service
FRESH	Pseudo-Random Number Generated by MS
GSM	Global System for Mobile communications
HE	Home Environment including HLR and AuC
HLR	Home Location Register
HMAC	Hash Message Authentication Code
HN	Home Network
IAKA	Identification and AKA
IK	Integrity Key
IMSI	International Mobile Subscriber Identity
KSI	Key Set Identifier
MAC	Message Authentication Code
MAC-I	Message Authentication Code for Integrity
MS	Mobile Station
MSC	Mobile Switching Center
$Nonce_X$	Pseudo-Random Number Generated by X
NV_X	Nonce Vector belongs to X
PIAKAP	Private Identification and AKA Protocol
PKD_X	Public-key Decryption with X's private key
PKE_X	Public-key Encryption with X's public key
PS	Packet Switched
RES	Response
RRC	Radio Resource Control
SE	Symmetric Encryption
SGSN	Serving GPRS Support Node
SKD	Symmetric-key Decryption
SKE	Symmetric-key Encryption
SMCM	Security Mode Complete Message
SN	Serving Network
SQN	Sequence Number
SRNC	Serving Radio Network Controller
TAK	Temporary Anonymity Key
TMSI	Temporary Mobile Subscriber Identity
UE	User Equipment
UEA	UMTS Encryption Algorithm
UIA	UMTS Integrity Algorithm
UMTS	Universal Mobile Telecommunications System
USIM	Universal Subscriber Identity Module
VLR	Visitor Location Register

II. INTRODUCTION

THE wireless communications advances cause the ease of access to wireless services for individuals. Since all the transmissions between the user and the network are sent over the air interface, the physical security of users' media is in serious danger compared to wired infrastructure. The methods which guarantee the security in physical layer like spread spectrum techniques are very expensive for commercial usages, so wireless providers try to secure higher network layers to enable privacy and confidentiality features for their subscribers.

The most commonly used wireless communications is cellular communications. In the first-generation (1G) of cellular networks, analog systems security services were not addressed at all which resulted to lots of security adventures. Proper

authentication of subscribers is an important feature for operators to charge them correctly. So in the second-generation (2G) digital cellular systems, GSM, users' confidentiality and authentication were taken into account in the design of GSM security measures.

However, GSM is still vulnerable to some security threats including active attacks performed by a fake BTS, insecure communications between a BTS and a BSC and also between a BSC and a MSC, lack of data integrity check, and weak stream cipher algorithm (A5/1,2) employed for providing confidentiality. These weaknesses were regarded in the security design phase of UMTS, a standard for the third-generation (3G) of mobile networks. As a result, an enhanced authentication and key agreement protocol and some security features like the integrity check as well as strong encryption algorithms were added to UMTS network.

Among the features mentioned above, the authentication and key agreement protocol has vital importance and the most prominent security aspects of the network are based on it. In the AKA protocol, besides authenticating each other, the user and the network agree on the cipher and integrity keys, CK and IK respectively. These keys are driven from the user's secret key (K) and revealing them may disclose some information about the user's secret key. Therefore if there exists any vulnerability in the AKA protocol, the subscriber's key (K) may be compromised. In other words, the user's confidentiality strongly depends on the security of the AKA protocol, and so great efforts should be taken to improve the security of AKA. We focus on this protocol for the purpose of improving its security and performance.

Our new protocol covers the stages of UMTS connection setup which consist of identification, AKA and security mode setup. With regarding to UMTS AKA we provide mutual authentication and fresh session keys generation with security mode setup. Moreover private identification, mutual fresh session keys derivation with reduced interactions and DoS attack resiliency is acquired properly. MS could anonymously roam in home or foreign networks and no one could trace the user. Only a local VLR could track MS for a temporary session.

This paper is organized as follows: in section III we explain the UMTS authentication and key agreement (AKA) procedure. Section IV outlines the related challenges to improve the security and performance of the UMTS AKA protocol. In section V our new protocol (PIAKAP) is proposed that rectifies the previous known weaknesses. Finally in section VI, security of PIAKAP is analyzed and evaluated.

III. UMTS AKA DESCRIPTION

The UMTS connection setup consists of three stages including the identification, AKA and security mode setup. At the identification stage, MS sends its identity to VLR/SGSN via SRNC. The purpose of UMTS AKA is to set up a mutual authentication as well as establish a new pair of cipher and integrity keys between VLR/SGSN and USIM [1]. The entities MS, VLR/SGSN, and HLR/AuC (HE) are involved in UMTS AKA protocol. At the security mode setup stage, MS responses to the command of VLR/SGSN which is about the preferred ciphering and integrity check algorithms.

The UMTS AKA process utilizes the secret key (K) and the cryptographic algorithms including f_1, f_1^*, f_2, f_3, f_4, f_5, and f_5^* shared between MS and HE. Furthermore, in order to check the freshness of the received messages and prevents the replay attack, HE and MS use the counters (SQN_{HE}) and (SQN_{MS}) respectively. These counters produce two sequences of numbers which will be compared with each other in each time of protocol execution. The authentication protocol is based on a secret key K which is unique for each user and resides only in the USIM and the database in HE. As illustrated in Figure1, the UMTS identification, distribution of authentication data, AKA, and security mode setup procedures work as follows.

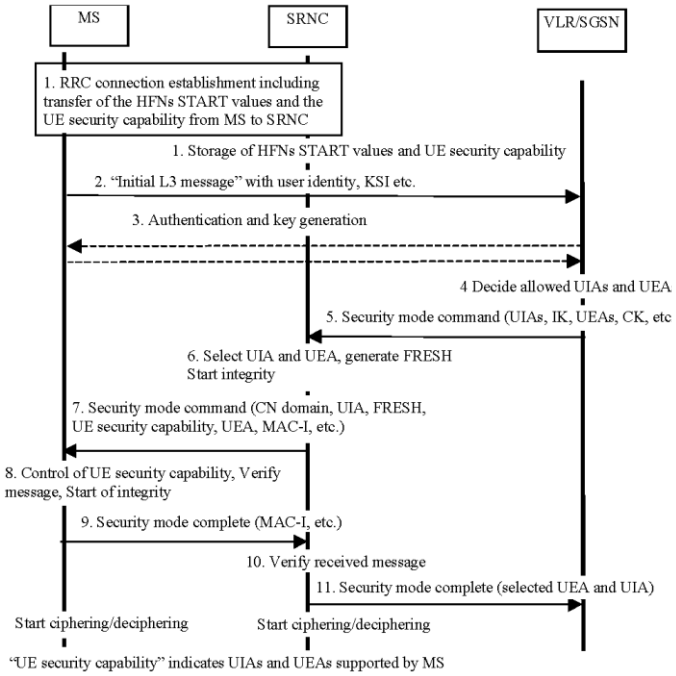

Fig. 1. The UMTS identification, distribution of authentication data, AKA, and security mode setup procedure [1]

Identification

The MS sends the security capabilities including UIA (UMTA Integrity Algorithm) and UEA (UMTS Encryption Algorithm) with START values of CS service domain to SRNC. Also the MS transfers the initial L3 message including its IMSI (or TMSI) to VLR/SGSN via SRNC. By this message, the MS requests for services like Location Update, CM (Connection Management) Service, and Routing Area Update.

Distribution of authentication data

The VLR/SGSN identifies the MS by its IMSI (or TMSI), and then sends the authentication data request including the MS's IMSI and the requesting node type (PS or CS) to the HE.

Upon the receipt of the authentication data request, the HE sends an authentication data response back to the VLR/SGSN that contains an ordered array of n authentication vectors AV(1...n). The generation of AV, which includes RAND, XRES, CK, IK, and AUTN is shown in Figure2.

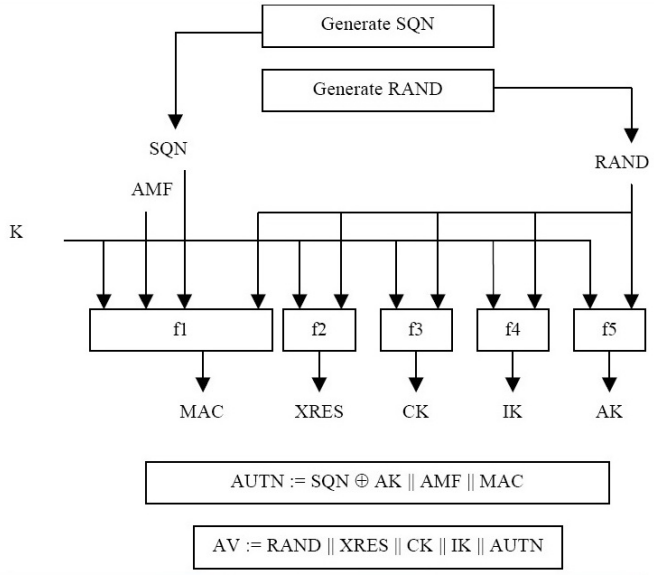

Fig. 2. Generation of Authentication Vectors [1]

Authentication and Key Management

The VLR/SGSN chooses the next unused AV from the ordered array of AVs in the VLR/SGSN database on the basis of first-in/first-out. Then, the VLR/SGSN sends to the MS a random challenge (RAND) and an authentication token (AUTN) from the chosen AV.

The operation of MS upon the receipt of RAND||AUTN is illustrated in Figure3. The MS first computes the anonymity key $AK=f5_K(RAND)$ and retrieves the sequence number. Then the MS computes $XMAC=f1_K(SQN||RAND||AMF)$ and compares it with MAC included in AUTN. If they are identical, the MS verifies if the SQN is in the correct range, compared to its sequence number. The MS calculates $RES=f2_K(RAND)$, if the SQN is considered to be in the correct range. Next, the MS sends RES to VLR/SGSN.

The VLR/SGSN compares the received RES with XRES. If they are equal, the AKA process of the MS is successfully completed.

Security Mode Setup

In this stage VLR/SGSN sends the security capabilities which are allowed to be used to SRNC as well as CK and IK. After that, SRNC selects the algorithms with the most priority compared to the user and network security capabilities. Then SRNC sends the selected algorithms, FRESH and MAC-I (which is generated by the SRNC) to MS. The last parameter is a message authentication code which is the result of employing the selected integrity

algorithm on the selected algorithms and user's security capabilities. For this purpose IK is used.

After receiving this message from SRNC, MS checks that the security capabilities are equal to the UEA and UIA sent in the initial message. The MS computes XMAC-I on the received message, and verifies the integrity of the message by comparing MAC-I with the computed XMAC-I. If all the checks are successful, the MS sends the RRC message security mode complete with MAC-I for it.

The SRNC by checking the integrity of the security mode complete message from the MS verifies the response of the MS and transfer the RANAP message security mode complete response containing chosen UIA and UEA to the VLR/SGSN.

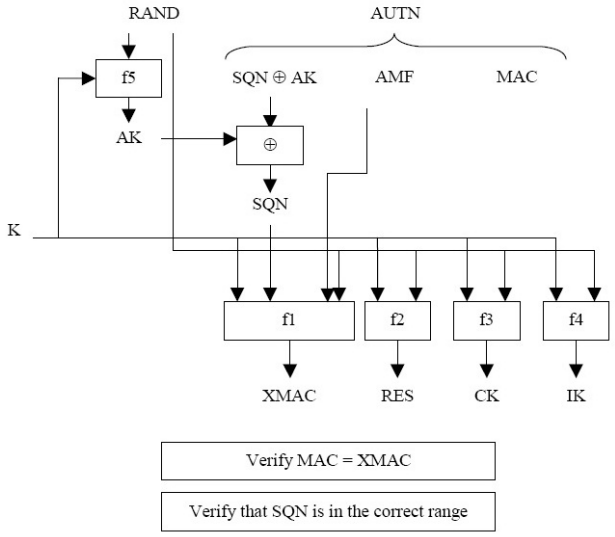

Fig. 3. User Authentication Function in the MS [1]

IV. Related Works Analysis

In this section we consider the security analysis of previous proposed improvements for UMTS AKA Protocols. Many protocols have been suggested for UMTS AKA improvement, but we chose some protocols that have true novelty in their design and use symmetric algorithms. Nevertheless they have some security or performance weaknesses in their structure that we try to explain.

The UMTS X-AKA protocol [7] applies a temporary key mechanism with timestamp instead of the sequence number. The function f5 is used for generating temporary keys. The UMTS X-AKA protocol consists of two procedures. First, the user registers with HN, and then HN distributes temporary key (TK) and authentication information to SN. Second, the authentication and key agreement procedure is performed between SN and MS. SN uses TK and information authentication to carry out the mutual authentication between SN and MS and then a cipher key and an integrity key are provided. The UMTS X-AKA protocol uses timestamp to manage freshness of the messages. The timestamp usage needs a time synchronization infrastructure. Time-sync structure of the network has no security feature, so the usage of an independent structure with no security to refresh the exchanged messages is hazardous. Also the HN could not recognize the shared session keys between MS and SN, because SN generates the pseudo-random number needed to construct the session keys.

In [8] an AKA protocol with robust user privacy protection has been proposed. This scheme not only uses a temporary key mechanism to authenticate MS and prevent the location privacy attack but it also makes lower overhead on VLR. Since MS can easily compute the temporary key through the shared secret key, VLR can be authenticated by MS successfully. In this protocol [8] the VLR initiates the authentication process by sending a nonce to the MS without any MAC, so DoS attack is probable to the MS. Also the protocol has seven steps without identification and security mode set-up stages.

J. Al-Saraireh and S. Yousef proposed an AKA protocol [9] that the MS generates the AVs sending to the network. They provided an efficient bandwidth consuming framework with minimal ways for the AKA procedure, but the proposed protocol doesn't support mutual authentication i.e. the network only authenticate the MS. The protocol has 3 steps. If the VLR/SGSN sent the RES received from the HE to the MS, the mutual authentication would be satisfied by checking the XRES and RES in the MS side. The man in the middle attack scenario on interworking of UMTS and GSM [10] could be applied on this protocol [9], because the MS doesn't recognize the validity of the network. Furthermore the DoS attack on the MS is possible, because the MS could only verify the network until a MAC is received from the network. If the VLR/SGSN sent the RES received from the HE to the MS, the mutual authentication would be satisfied by checking the XRES and RES

in the MS-side.

The security of the wireless network access has been enhanced by Harn and Hsin [11]. Their scheme benefits from timestamp and hash chain to provide non-repudiation and freshness. As we mentioned earlier, using the timestamp need independent secure infrastructure, also the hash chain construction necessitates high processing load at the end users side. Furthermore the number of the protocol rounds is six and the protocol doesn't contain the identification and security mode set-up stages.

An extension of UMTS AKA protocol has been proposed by J. Al-Saraireh and S. Yousef [12] that provides mutual freshness of the MS and the HE, generating a nonce number by both entities, but it doesn't use sequence number mechanism. So for preventing replay attack, verifying the freshness of the messages could be done by searching in a large database that contains all of the previous nonce generated by the parties. Applying such a huge database is more expensive and complex than using sequence number method. Besides the power of DoS attack diminishes when the parties can check the integrity and freshness of the messages faster.

M. Zhang and Y. Fang analyzed and enhanced the security of the 3GPP AKA by a protocol called AP-AKA. They have projected a special scenario of the redirection attack that UMTS AKA is weak towards it and AP-AKA is robust against it. But the MS traffic redirection by a virtual relay to the neighbor VLR could charge the MS more than usual because the location of the MS has been virtually changed. The first step of the AP-AKA has not integrity protected, so it could be forged. Also the attack [10] can be executed on AP-AKA while interworking with GSM, because the VLR initiates the AKA procedure without integrity check. Furthermore the identification and security mode set-up stages are not considered to provide a better performance for the AP-AKA six-round protocol.

V. Private Identification, Authentication and Key Agreement Protocol

In this section we explain the stages of a new proposed authentication protocol of UMTS mobile networks. As described in previous section identification, authentication and key agreement with security mode set-up protocol of UMTS networks has some weaknesses to provide simple DoS-attack resistance. Also proposed schemes do not provide mutual freshness i.e. the generation of the random numbers by the MS and the HE. Furthermore the steps of the pre-proposed protocols could be reduced to save bandwidth consumption.

The new private identification and AKA protocol, named PIAKAP, for UMTS mobile network combines identification stage and AKA stage with security mode setup of described UMTS AKA protocol. A five-way handshake protocol between MS, VLR/SGSN, and HE is used to establish secure connection in full version of PIAKAP. After first identification PIAKAP could be performed by three interactions between MS and VLR/SGSN. Most of the previous schemes were done by a 5-way handshake in the phase of AKA without security mode set-up, so we could enhance bandwidth efficiency.

PIAKAP applies the secret key (K) and the cryptographic algorithms that are used in UMTS AKA protocol shared between MS and HE. But the usage of the f2 algorithm in PIAKAP is not necessary because PIAKAP does not generate RES or XRES values.

Both the MS and the HE have sequence number and random number generator to provide complete freshness of their messages with nonce numbers. On the basis of the f5 structure [2] we could improve the confidentiality of the SQN by changing its encryption method.

To protect VLR/SGSN against DoS attack, we assume some security capabilities at VLR/SGSN side. These security features contain the shared cryptographic algorithms (SKE and HMAC) between MS and VLR/SGSN. The SKE algorithm is used for ciphering and the HMAC algorithm is used to generate integrity check. PIAKAP procedure while MS meets a new VLR/SGSN is shown in Figure4. Also Figure5 depicts the steps of PIAKAP, when MS has a TMSI allocated by an old visited VLR/SGSN. PIAKAP steps works as follows:

Authentication Request

1- If no TMSI was allocated to the MS by the VLR/SGSN, the MS generates AREQ message as follows. "+" sign is bit-wise XOR operator with the

size of nonce number.

$AREQ=Cert_{HE}||CHALLENGE_{HE}||MAC_{MS,AK}$

Where

$CHALLENGE_{HE}=PKE_{HE}(NV_{MS}(1..n),SQN_{MS},IMSI)$

$AK=f3_K(Nonce_{MS,1}+\ldots+Nonce_{MS,n})$

$MAC_{MS,AK}=f1_{AK}(SQN_{MS}||IMSI||Cert_{HE})$

The nonce vector field contains some pseudo-random numbers generated by the MS and the SQN_{MS} refers to the first generated sequence number of AREQ. Because the RSA public key block is at least 1024-bit, the MS could embed enough 128-bit nonce numbers in $CHALLENGE_{MS}$ field. The number of embedded nonce numbers is maintained in the MS's counter, and the counter is decreased every successful authentication by one. If successful authentication requests reach the size of the nonce vector (NV), the counter value is zero and a new AREQ must be produced like above.

If the VLR/SGSN allocated a TMSI encrypted by previous CK (OCK) paired with previous IK (OIK) to the MS, AREQ message would be generated by the MS as follows:

$AREQ=Cert_{VLR}||CHALLENGE_{VLR}||MAC_{MS,OIK}$

Where

$CHALLENGE_{VLR}=TMSI||SKE_{OCK}(Nonce_{MS},SQN_{MS})$

$MAC_{MS,OIK}=HMAC_{OIK}(Nonce_{MS}||SQN_{MS}||TMSI||Cert_{VLR})$

Then the MS sends the computed AREQ to the VLR/SGSN via the SRNC.

Distribution of authentication data

2- The VLR/SGSN checks the sent certificate. If the HE's certificate is attached, the AREQ message is directly forwarded to the HE. If the MS attached the identity of the pre-visited VLR/SGSN, the message is processed by the proper VLR/SGSN. The intended VLR/SGSN verify the MS's TMSI and uses the last shared cipher key between the MS and itself to decrypt the ciphered section of the $CHALLENGE_{VLR}$ parameter and get the MS's $Nonce_{MS}$ and SQN_{MS}.

If the MS's TMSI related to current VLR/SGSN is sent, the VLR/SGSN computes the $XMAC_{VLR,OIK}$ (as given below) by the challenge parameters, and compares this with $XMAC_{MS,OIK}$ which is included in AREQ.

$XMAC_{VLR,OIK}=HMAC_{OIK}(Nonce_{MS}||SQN_{MS}||TMSI||Cert_{VLR})$

If they are different, the authentication process is abandoned by the VLR/SGSN. Otherwise if the VLR/SGSN has some authentication vectors, it will initiate step4 of this procedure. If there is no authentication vector related to the MS in the VLR/SGSN, the MS identification is initiated as described in step1.

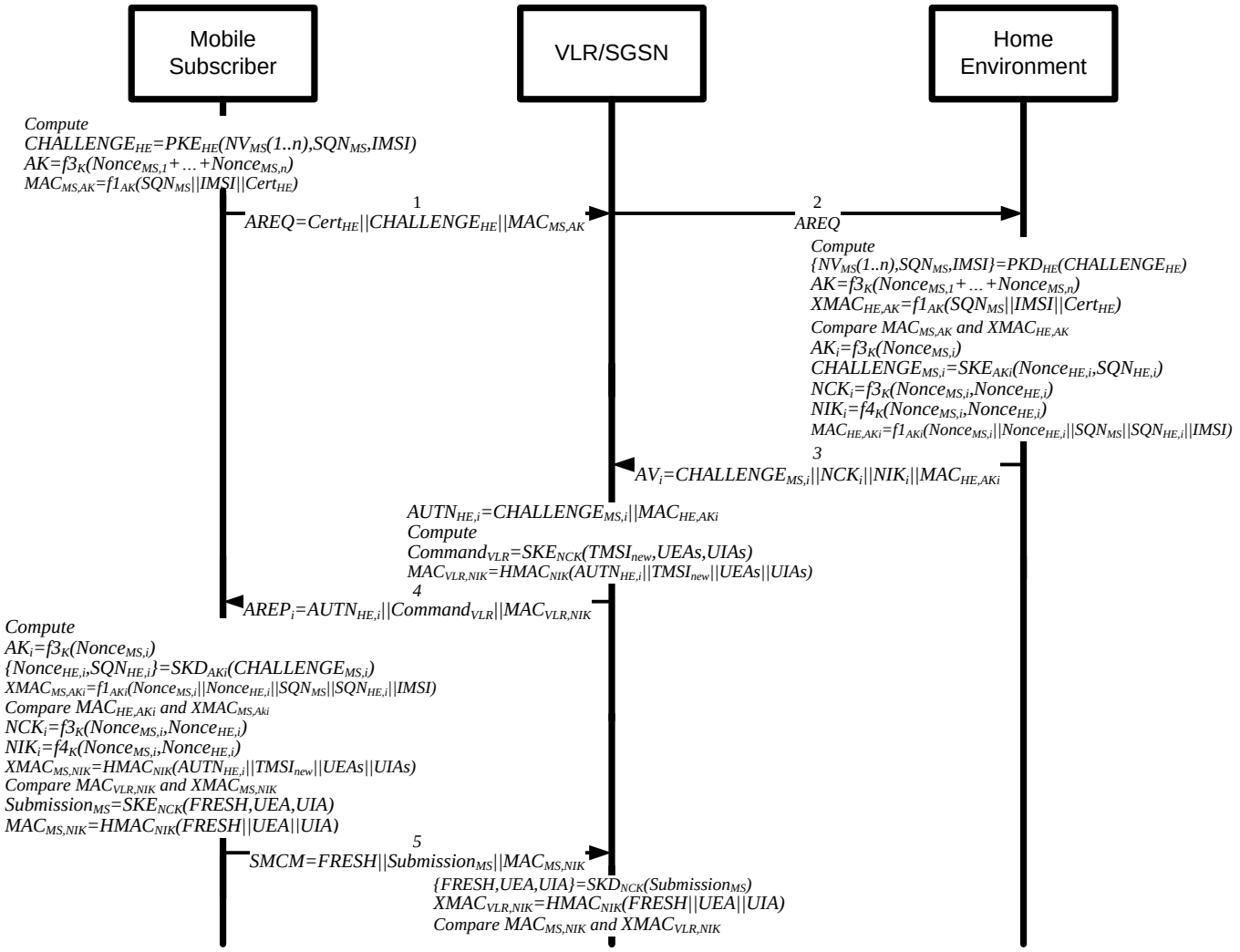

Fig. 4. PIAKAP Structure when MS meets a new VLR/SGSN

3-When the HE receives the authentication data request from the VLR/SGSN, it first retrieves the MS's identity, sequence number, and nonce vector as follows.

$\{NV_{MS},SQN_{MS},IMSI\}=PKD_{HE}(CHALLENGE_{HE})$

Then the HE computes the $XMAC_{HE,AK}$ (as given below), and compares it with $MAC_{MS,AK}$ which is included in the VLR/SGSN's authentication data request.

$AK=f3_K(Nonce_{MS,1}+\ldots+Nonce_{MS,n})$

$XMAC_{HE,AK}=f1_{AK}(SQN_{MS}||IMSI||Cert_{HE})$

If the $MAC_{MS,AK}$ and the $XMAC_{HE,AK}$ are different, the HE detects the MS as a fraud user and does not generate any AV. If they are equal, the HE verifies the SQN_{MS} is in the correct range.

If the SQN_{MS} is considered to be in the correct range, The HE sends an authentication data response back to the VLR/SGSN that contains some authentication vectors (AVs) with SQN_{MS}. Every generated AV_i including $Nonce_{HE,i}$, new CK (NCK_i), new IK (NIK_i), and $MAC_{HE,K,i}$.

For i=1…n

$AV_i=CHALLENGE_{MS,i}||NCK_i||NIK_i||MAC_{HE,AKi}$

Where

$AK_i=f3_K(Nonce_{MS,i})$

$CHALLENGE_{MS,i}=SKE_{AKi}(Nonce_{HE,i},SQN_{HE,i})$

$NCK_i=f3_K(Nonce_{MS,i},Nonce_{HE,i})$

$NIK_i=f4_K(Nonce_{MS,i},Nonce_{HE,i})$

$MAC_{HE,AKi}=f1_{AKi}(Nonce_{MS,i}||Nonce_{HE,i}||SQN_{MS}||SQN_{HE,i}||IMSI)$

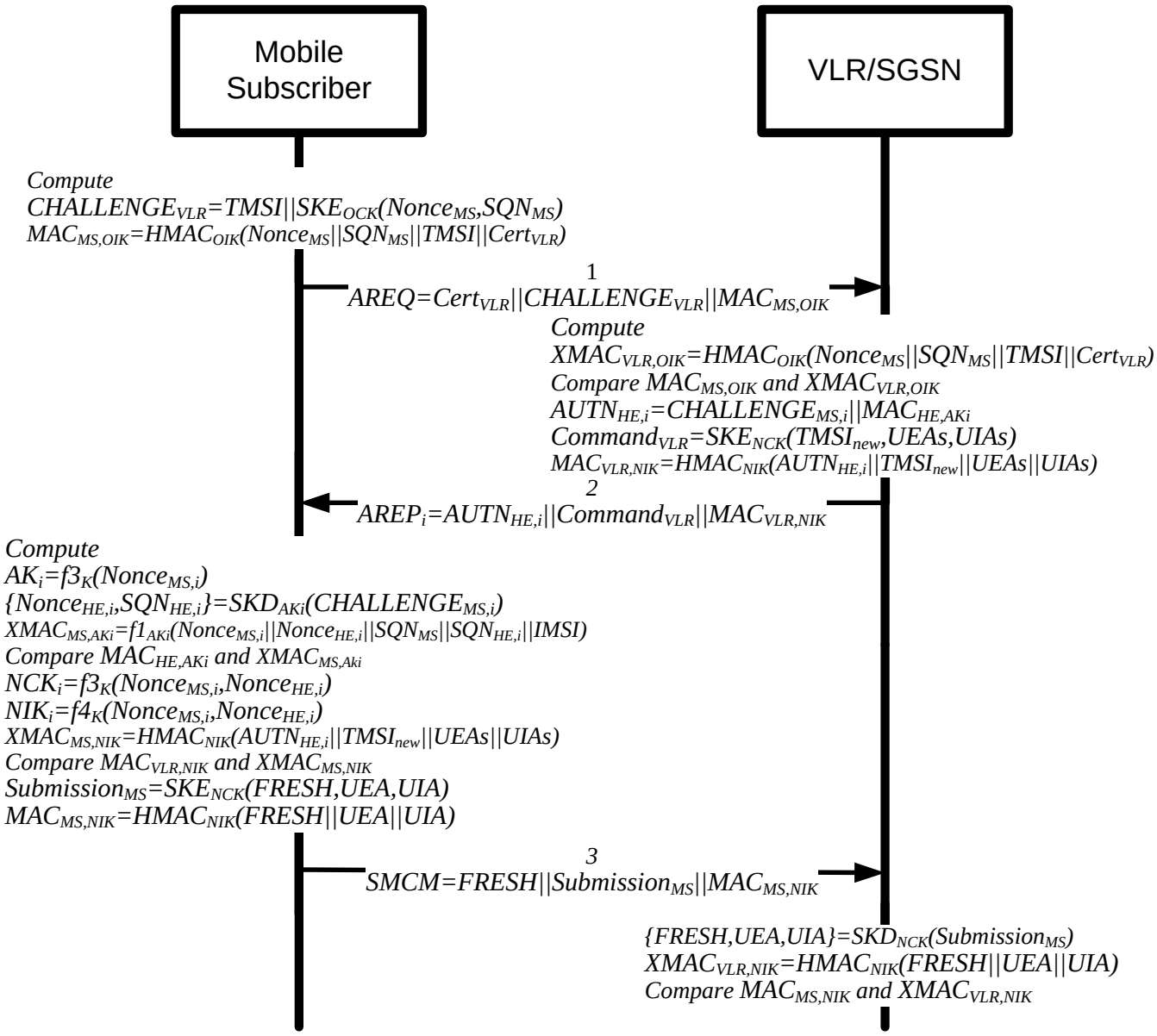

Fig. 5. PIAKAP Structure when MS meets an old visited VLR/SGSN

Authentication Reply

4- Upon the receipt of the AVs from the HE, the VLR/SGSN computes the AREP for the MS. A new TMSI is generated for the MS. The VLR/SGSN determines which UIAs and UEAs that are allowed to be used in order of preference. The VLR/SGSN generates the AREP message as follows:

$AREP_i=AUTN_{HE,i}||Command_{VLR}||MAC_{VLR,NIK}$

Where

$AUTN_{HE,i}=CHALLENGE_{MS,i}||MAC_{HE,AKi}$

$Command_{VLR}=SKE_{NCK}(TMSI_{new},UEAs,UIAs)$

$MAC_{VLR,NIK}=HMAC_{NIK}(AUTN_{HE,i}||TMSI_{new}||UEAs||UIAs)$

5- Upon the receipt of the AREP message, the MS proceeds as illustrated in Figure4/5. the MS first derives i-th anonymity key (AK_i) from its $Nonce_{MS,I}$ and K, then retrieves the HE's nonce and sequence number as follows:

$AK_i=f3_K(Nonce_{MS,i})$

$\{Nonce_{HE,i},SQN_{HE,i}\}=SKD_{AKi}(CHALLENGE_{MS,i})$

If the SQN_{HE} is considered to be in the correct range, the MS computes $XMAC_{MS,AK,i}$ (as given below) and compares it with the $MAC_{HE,AK,i}$.

$XMAC_{MS,AKi}=f1_{AKi}(Nonce_{MS,i}||Nonce_{HE,i}||SQN_{MS}||SQN_{HE,i}||IMSI)$

If they are the same, the MS calculates the new pair key, the new cipher key (NCK) and the new integrity

key (NIK), as follows:

$NCK_i=f3_K(Nonce_{MS,i},Nonce_{HE,i})$

$NIK_i=f4_K(Nonce_{MS,i},Nonce_{HE,i})$

Then the MS decrypts the command of the VLR/SGSN and gets its new ordered TMSI and the cryptography capabilities of the VLR/SGSN in order of preference. Then the MS computes $XMAC_{MS,NIK}$ (as given below) with regarding to the allocated TMSI.

$$XMAC_{MS,NIK}=HMAC_{NIK}(AUTN_{HE,i}||TMSI_{new}||UEAs||UIAs)$$

The MS compares $XMAC_{MS,NIK}$ with $MAC_{VLR,NIK}$ included in the received $AREP_i$ message. If they are different, the MS does not continue authentication process and terminate the connection. Otherwise the MS generates the security mode complete message (SMCM) as follows:

$SMCM=FRESH||Submission_{MS}||MAC_{MS,NIK}$

Where

$Submission_{MS}=SKE_{NCK}(FRESH,UEA,UIA)$

$MAC_{MS,NIK}=HMAC_{NIK}(FRESH||UEA||UIA)$

The MS sends the SMCM to the VLR/SGSN, and the VLR/SGSN decrypts the $Submission_{MS}$ by the new shared ciphering key (NCK) to get preferred UEA and UIA.

$\{FRESH,UEA,UIA\}=SKD_{NCK}(Submission_{MS})$

Then the VLR/SGSN verifies the integrity of the SMCM by generating the $XMAC_{VLR,NIK}$ with regarding to the chosen UEA and UIA by the MS and comparing it with the $MAC_{MS,NIK}$.

$XMAC_{VLR,NIK}=HMAC_{NIK}(FRESH||UEA||UIA)$

Thus IAKA process with security mode setup is complete i.e. the parties authenticate each other mutually and preferred algorithms for ciphering and integrity check are chosen by the parties.

VI. Security and Performance of PIAKAP

In this section we describe the security features of the new proposed protocol explained in previous section and then we evaluate the performance of the suggested protocol relatively. PIAKAP provides the main security services issued in literature like authentication, confidentiality and integrity [3]. Furthermore the PIAKAP is more robust than the UMTS AKA and most of the preceding suggested protocols against the DoS attack, so with using the PIAKAP, availability service would be provided.

A. Confidentiality and Integrity

As described in previous section, the only shared secret between MS and HE is the private key (K). Also, an old pair of session keys (OCK, OIK) may be shared between MS and VLR/SGSN. if MS has already been in VLR/SGSN domain. In any case, they would share a new session pair key (NCK, NIK) through the PIAKAP. These keys are hidden from the sight of adversary, and we show that they could not be guessed or eavesdropped easily.

Nonce fields as random numbers needed for the other involved entity to generate new shared keys are encrypted. Moreover, the sequence numbers, UEA, and UIA fields are encrypted by the shared keys (AK, AK_i or NCK). Therefore no third party could perceive even stream-ciphering and integrity check algorithm used for secure connection. Moreover the confidentiality service is provided properly. Also MAC fields are protected by the shared keys (AK, AK_i, OIK, or NIK). Consequently the integrity service is afforded appropriately.

B. Anonymity vs. Authentication

The identification fields are sent ciphered to provide anonymity service. VLR/SGSN allocates a new TMSI to the MS whenever the protocol executes. When MS visits a new VLR/SGSN for the first time, they mutually authenticate each other via HE. For the first identification to new VLR/SGSN, MS uses public-key encryption to covertly send its identity to the HE. Also this method helps MS to hide its identity from any stranger VLR when MS roams in a foreign network. Local VLR/SGSN could only trace MS; While MS has established a temporary session with current local VLR in its region.

The identity of MS could be authenticated at HE-side by verifying the integrity of MS's generated $MAC_{MS,AK}$. MS could confirm the identity of its home network by checking the $MAC_{HE,AKi}$ calculated by the HE. So two parties identify each other suitably i.e. the mutual identification is accomplished.

Besides MS and VLR/SGSN could authenticate each other by verifying the $MAC_{MS,OIK}$, $MAC_{MS,NIK}$ and $MAC_{VLR,NIK}$, as the session keys (OIK, NIK) could be computed or available in both sides, So the mutual authentication with security mode set-up is done by PIAKAP.

C. Sequence Number Protection vs. Private Key Derivation

MS increments the SQN_{MS} for every PIAKAP execution and HE runs the SQN_{HE} for every generated AV. MS and HE used sequence number and random number generator to maintain strong freshness of their messages. Although pseudo-random numbers could provide the freshness of the ways of PIAKAP, both MS and HE need a large directory to save used random numbers in it. Also to avoid the usage of the repeated random numbers, they have to search in their directory or sort and update the directory for every procedure which it consumes a large number of energy and memory.

In PIAKAP, anonymity protection of the sequence number is higher than the UMTS AKA. Because the sequence number of the UMTS AKA is XOR-ed by anonymity key derived from nonce and the private key [2] like stream ciphering. As sequence number value is guessable because it starts from zero and changes incrementally slowly so the known plaintext attack is imaginable on the f5 algorithm to get private key (K). However the core of the f5 algorithm is based on Advanced Encryption Standard (AES) [4] that the recent proposed attacks are not practical on it [5].

The sequence number encrypts in PIAKAP by a block cipher called *SKE* algorithm with derived key from f3 algorithm, so the privacy of the sequence number is strongly established.

D. Availability (Robustness against DoS Attack)

If MS enters a new area (A1) of VLR/SGSN (V1), it won't have a TMSI allocated by VLR/SGSN (V1). So MS must use public-key encrypted IMSI to initiate the procedure. In this situation MS actually aim its HE not VLR/SGSN (V1). In fact SRNC (SR1) and VLR/SGSN (V1) in A1 role as relays to forward MS messages to HE.

First we consider the condition that no TMSI was set before. MS and VLR/SGSN have no shared key before so as mentioned in step1 of the previous section AREQ message has been sent to HE via intermediate SRNC and VLR/SGSN. As mentioned in step3, HE could check the integrity of the $MAC_{MS,AK}$, so the spam messages generated by spurious users are discarded by HE. Furthermore the DoS attack organized by unauthentic MS at the step3 could be detected at HE side, and no more traffic would be procreated in the network.

Second we consider the situation where the VLR/SGSN allocated MS a TMSI, so MS and VLR/SGSN have shared a pair key (OCK and OIK) with each other. Consequently as explained in step2, the VLR/SGSN checks the integrity of the $MAC_{MS,OIK}$, and rejects the forged messages. Furthermore with the usage of MS resume, the network could prevent DoS attack of first step at VLR/SGSN side.

In the other steps of the PIAKAP by using MAC mechanism the vulnerability towards DoS attack is reduced like as mentioned above. Although in some cases the SQN should be computed before the MAC, so the computation load would be increased. But no further signaling load would be injected to the network and the elements of the network could decide about the genuineness of the message by verifying the MAC integrity.

E. Mutual freshness and unguessable keys

The SQN_{MS} is encrypted by anonymity key (AK) derived from the output of the f3 algorithm by a high entropy seed, $Nonce_{MS,1}+\ldots+Nonce_{MS,n}$. The $SQN_{HE,i}$ is protected by anonymity key (AKi) derived from the f3 algorithm by a random seed $Nonce_{MS,i}$, that is an entropic seed to generate a fresh key. The ciphered SQN_{HE} is dependent on two random numbers, $Nonce_{MS}$ and $Nonce_{HE,i}$, is high enough to be guessed by adversary. So the sequence numbers could not be revealed. Also if an anonymity key compromise at a session, no next-generated keys will be concealed i.e. our scheme supports forward security towards session key compromising.

The MAC fields like the SQN fields are fresh. The entropic seed of the MAC fields is derived from the pseudo random numbers as well IMSI and sequence number. The session *freshness entropy (Fr)* for a message (M) or a field of a message is defined recursively with the assumption of perfect secure algorithms usage as follows:

$Fr(M):=Fr(Seed(M_{new}|M_{olds})$

Where the initialization values are as follows:

$Fr(Nonce)=H(Nonce)=log_2||Nonce||$

$Fr(SQN)=H(SQN_{new}|SQN_{olds})=log_2||\Delta SQN||$

$Fr(K)=H(K)=log_2||K||$

$Fr(IMSI)=H(IMSI)$

Where H(A|B) denotes the equivocation of random variable A subject to the random variable B [6], ||.|| denotes the operator that get the number of bits, and Seed(M_{new}) means the material needed to construct a new message (M) like Nonces, SQNs, IMSI, and K. The M_{olds} refers to all previously generated messages that the generation procedure of is like the M_{new} one. The ΔSQN shows the variance of the sequence number growth. When no TMSI was set before for the MS, we could get the freshness entropy of the messages with regarding that nonce numbers and SQNs generate independently and the length of AV is one:

$$Fr(AREQ)=Fr(Seed(AREQ_{new})|AREQ_{olds})$$
$$=H(K,IMSI,newNonce_{MS},newSQN_{MS}$$
$$|IMSI,oldNonces_{MS},oldSQNs_{MS})$$
$$=H(newNonce_{MS})+H(\Delta SQN_{MS})$$

$$Fr(CHALLENGE_{HE})|AREQ_{olds})$$
$$=H(IMSI,newNonce_{MS}|AREQ_{olds})$$
$$+H(newSQN_{MS}|oldSQNs_{MS})=Fr(AREQ)$$

$$Fr(MAC_{MS,AK}|AREQ_{olds})$$
$$=H(K,IMSI,newNonce_{MS},newSQN_{MS}|AREQ_{olds})$$
$$=Fr(AREQ)$$

Consequently the freshness of the AREQ is equal with its protected components like concealed sequence number and message authentication code (MAC). When a TMSI was set before for the MS, we could get the below deduction as well.

$$Fr(AREQ)$$
$$=Fr(CHALLENGE_{HE}|AREQ)$$
$$=Fr(MAC_{MS,K}|AREQ)$$

The freshness entropies of the other fields including the new pair key (NCK and NIK) of the PIAKAP are as below:

$$Fr(NCK|PIAKAP_{olds})$$
$$=Fr(K,IMSI,Nonce_{MS},Nonce_{HE}|PIAKAP_{olds})$$
$$=H(Nonce_{MS})+H(Nonce_{HE})$$
$$Fr(NIK|PIAKAP_{olds})$$
$$=Fr(K,IMSI,Nonce_{MS},Nonce_{HE}|PIAKAP_{olds})$$
$$=H(Nonce_{MS})+H(Nonce_{HE})$$

The maximum freshness of the protocol is exist in the generation of the new pair key, so the reply attack probability is about $2^{-Fr(NCK|PIAKAP)}$ that is so small.

F. Performance towards Previous Works

The PIAKAP combined three section of the UMTS protocol to set-up connection called identification, AKA, and security mode set-up. In PIAKAP security mode setup is after key establishment, so the attacker has no information about the shared algorithms for ciphering and integrity check. PIAKAP completes after five-way interactions at most. When a TMSI allocated for MS, PIAKAP accomplishes after three-way handshake between MS and VLR/SGSN. Previous works do not consider the identification and security mode set-up stages. Furthermore if we consider the whole protocol, the number of interactions has been reduced largely. So the load of the network signaling is reduced, and performance of the PIAKAP procedure efficiently grows.

VII. CONCLUSION

Most of the proposed protocols do not consider the previous and next stages of the UMTS AKA protocol and they try to improve the UMTS AKA protocol solely. Our proposed protocol, so called PIAKAP, merges all of the stages in five ways, so it improves performance by reducing the load of the network signaling. Also PIAKAP support mutual freshness of agreed keys and is more robust against DoS attack by applying MAC mechanism between MS and VLR/HE.

REFERENCES

[1] 3GPP TS 33.102 V8.0.0 (2008-06), 3GPP Technical Specification Group Services and System Aspects, 3G Security, Security Architecture (Release 8).

[2] 3GPP TS 35.206 V7.0.0 (2007-06), Technical Specification Group Services and System Aspects, 3G Security, Algorithm Specification (Release 7).

[3] William Stallings, Cryptography and Network Security: Principles and Practice, Prentice-Hall, July 1998, ISBN 0138690170.

[4] National Institute of Standards and Technology (NIST), Advanced Encryption Standard (AES), Federal Information Processing Standards Publications (FIPS PUBS) 197 (2001).

[5] H. Dobbertin, L. Knudsen, and M. Robshaw, "The cryptanalysis of the AES - a brief survey", Advanced Encryption Standard C AES: 4th International Conference, AES 2004, volume 3373 of Lecture Notes in Computer Science, pp. 1–10, Springer-Verlag, 2005.

[6] C. E. Shannon, "A Mathematical Theory of Communication", the Bell System Technical Journal, Vol. 27, pp. 379–423, 623–656, July, October, 1948.
[7] C.M. Huang and J.W. Li, “Authentication and Key Agreement Protocol for UMTS with Low Bandwidth Consumption”, Proceedings of the 19th International Conference on Advanced Information Networking and Applications, 2005.
[8] W.S. Juang and J.L. Wu, "Efficient 3GPP Authentication and Key Agreement with Robust User Privacy Protection", IEEE Communications Society, Proceedings of the WCNC, 2007.
[9] J. Al-Saraireh and S. Yousef, "A New Authentication Protocol for UMTSMobile Networks", EURASIP Journal on Wireless Communications and Networking, Article ID 98107, pp. 1–10, 2006.
[10] U. Meyer and S. Wetzel, "A Man-in-the-Middle Attack on UMTS", Proceedings of the 3rd ACM workshop on Wireless security, pp. 90-97, ISBN:1-58113-925-X, 2004.
[11] L. Harn and W.J. Hsin, "On the Security of Wireless Network Access with Enhancements", Proceedings of the 2nd ACM workshop on Wireless security, pp. 88-95, ISBN:1-58113-769-9, 2003.
[12] J. AL-Saraireh and S. Yousef, "Extension of Authentication and Key Agreement Protocol (AKA) for Universal Mobile Telecommunication System (UMTS)", International Journal of Theoretical and Applied Computer Sciences, Vol. 1, No. 1, pp. 109–118, 2006.
[13] M. Zhang and Y. Fang, "Security Analysis and Enhancements of 3GPP: Authentication and Key Agreement Protocol", IEEE Transactions on Wireless Communications, Vol. 4, No. 2, March 2005.
[14] Farhat, Farshid, Mohammad Heydari, and Mohammad Reza Aref. "Security Weaknesses in PGP Protocol."
[15] Farhat, Farshid, et al. "Eigenvalues-based LSB steganalysis." The ISC International Journal of Information Security 4.2 (2012): 97-106.
[16] Farhat, Farshid, Somayeh Salimi, and Ahmad Salahi. "An extended authentication and key agreement protocol of UMTS." International Conference on Information Security Practice and Experience. Springer Berlin Heidelberg, 2009.
[17] Farhat, Farshid, et al. "Locally multipath adaptive routing protocol resilient to selfishness and wormholes." International Conference on Information Security Practice and Experience. Springer Berlin Heidelberg, 2010.
[18] Tootaghaj, Diman Zad, et al. "Risk of attack coefficient effect on availability of Ad-hoc networks." Consumer Communications and Networking Conference (CCNC), 2011 IEEE. IEEE, 2011.
[19] Tootaghaj, Diman Zad, et al. "Game-theoretic approach to mitigate packet dropping in wireless Ad-hoc networks." Consumer Communications and Networking Conference (CCNC), 2011 IEEE. IEEE, 2011.
[20] Farhat, Farshid, Somayeh Salimi, and Ahmad Salahi. "Private Identification, Authentication and Key Agreement Protocol with Security Mode Setup." IACR Cryptology ePrint Archive 2011 (2011): 45.
[21] Farhat, Farshid, et al. "Multi-dimensional correlation steganalysis." Multimedia Signal Processing (MMSP), 2011 IEEE 13th International Workshop on. IEEE, 2011.
[22] Diyanat, Abolfazl, Farshid Farhat, and Shahrokh Ghaemmaghami. "Image steganalysis based on SVD and noise estimation: Improve sensitivity to spatial LSB embedding families." TENCON 2011-2011 IEEE Region 10 Conference. IEEE, 2011.