2009 Third IEEE International Conference on Secure Software Integration and Reliability Improvement

Provable Secure AKA Scheme with Reliable Key Delegation in UMTS

Yu-Lun Huang
Dept of Electrical Engineering
National Chiao Tung University
Hsinchu, Taiwan
E-mail: ylhuang@cn.nctu.edu.tw

C. Y. Shen, Shiuhpyng Shieh
Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan
E-mail: ssp@cs.nctu.edu.tw

Hung-Jui Wang, Cheng-Chun Lin
Institute for Information Industry
Taipei, Taiwan

***Abstract*— The Authentication Key Agreement Scheme (AKA) of Universal Mobile Telecommunication System (UMTS) provides substantial enhancement to solve the vulnerabilities in GSM and other wireless communication systems. However, we discovered four security weaknesses of UMTS AKA, that is, redirection attack, man-in-the-middle attack, sequence number depletion, and roaming attack. An adversary can launch these attacks to eavesdrop, or cause billing problems. To cope with these problems, a new Secure Authentication Key Agreement Protocol (S-AKA) is proposed in this paper to enhance the security to resist the attacks. To improve the efficiency and redundancy of UMTS AKA, S-AKA reduces both the authentication messages and bandwidth consumption of UMTS AKA. The formal proof of S-AKA is also given to ensure the security strength of S-AKA.**

Keywords- authentication, UMTS AKA, Man-in-the-middle attack, redirection attack.

I. INTRODUCTION

With the fast growth of cellular phone coverage, more and more mobile applications are developed and deployed. Nowadays the third generation (3G) mobile phones **[1]** are used widely together with its predecessor, the second generation (2G) mobile phones, also known as Global System for Mobile (GSM) mobile phones **[2]**. The goals of 3G mobile systems are to enhance service capabilities, provide global roaming operations, and improve the performance of the entire network. From the security perspective, 3G mobile systems intend to reduce or even eliminate the drawbacks of the second-generation (2G) mobile systems, which include: 1) only unidirectional authentication is provided, which may lead to the false base station attack, 2) authentication triplets can be reused, and 3) weak encryption is employed. Among the 3G mobile telecommunications technologies, the Universal Mobile Telecommunications System (UMTS) **[1]** is probably the most popular one. To address the security weaknesses in GSM, UMTS has adopted an enhanced authentication and key agreement protocol, called UMTS AKA. UMTS AKA achieves higher security level, that is, 1) mutual authentication between the mobile station (MS) and the serving network (SN), 2) agreement on an integrity key (IK) between the MS and the SN, and 3) freshness assurance of the mutually agreed cipher key (CK) and IK. The security enhancement in UMTS AKA resolved most of the vulnerabilities discovered in the GSM systems, and made UMTS a more secure telecommunication system **[3]**. However, UMTS AKA is still vulnerable to some attacks, including redirection attack **[4]**, man-in-the-middle attack **[11]**, sequence number depletion attack, and roaming attack. Under these attacks, victim users may be mischarged or even eavesdropped. Some researches [5][6][7][8][9][14] intend to improve the security of UMTS AKA, but their schemes still cannot resist aforementioned attacks. Our proposed scheme is aimed to eliminate the vulnerabilities, and to enhance the efficiency. We also provide the efficiency analysis of both UMTS AKA and the proposed scheme, S-AKA. The formal proof of S-AKA shows the security strength of S-AKA.

The rest of this paper is organized as follows. Section 2 introduces UMTS AKA and describes its security and bandwidth drawbacks. In Sections 3 and 4, we propose a new scheme S-AKA, analyze and compare its security and bandwidth with UMTS AKA. Te security of S-AKA is formally proved in Section 5, and finally Section 6 concludes the paper.

II. OVERVIEW OF UMTS AKA

In this section, UMTS AKA will be briefly introduced. In UMTS AKA **[1]**, three entities are involved, namely, a mobile station (MS), Serving GPRS Support Node (SGSN), and Home Location Register/Authentication Center (HLR/AuC). The MS acts on behalf of the user to communicate with the SGSN and HLR/AuC for mutual authentication. The SGSN represents the SN, which the MS visits, and the HLR/AuC in the home domain is in charge of the authentication data management. The MS and HLR/AuC share a secret key K, and some cryptographic functions, including f1, f1*, f2, f3, f4, f5, and f5*. Functions f1 and f1* are message authentication functions used to compute Message Authentication Code (MAC); function f2 is for computing response (RES) and expected response (XRES); functions f3, f4, f5, and f5* are key generation functions used to compute CK, IK, AK in the normal procedures, and AK in re-synchronization procedures, respectively. Each MS and HLR/AuC maintains a sequence number, SQN_{MS} and SQN_{HN}, to fight against replay attack.

The UMTS AKA authentication messages exchanged are given below. Each message is denoted as M_i and will be analyzed later in this paper.

978-0-7695-3758-0/09 $25.00 © 2009 IEEE
DOI 10.1109/SSIRI.2009.62

UMTS AKA authentication messages:

M1 MS sends a registration request containing its permanent International Mobile Subscriber Identity (IMSI) to the SGSN.

M2 Visited SGSN passes the registration request containing IMSI to HLR/AuC.

M3 HLR/AuC sends an ordered array of *n* authentication vectors AV(i) to the SGSN, where *i=1, ..., n*. Each AV(i) consists of a random number RAND(i), XRES(i), CK(i), IK(i) and an authentication token AUTN(i).

M4 SGSN selects the next unused AV(i) from the ordered array and sends RAND(i) and AUTN(i) to the MS.

M5 The MS checks whether AUTN(i) can be accepted or not. If accepted, the MS produces a response RES(i) and send it back to SGSN. The MS also computes the CK(i) and IK(i). SGSN compares the received RES(i) with stored XRES(i). If matched, the MS is authenticated and the procedure is successfully completed. Otherwise, the MS is denied.

Upon receipt of the fourth message M4, the MS authenticates the SN by checking if the MAC in the AUTN(i) is correct. The MS further verifies whether the sequence number in the AUTN(i) is in the correct range or not. If true, MS successfully authenticates the SNm. In message M5, The MS sends RES(i) to SGSN. SGSN checks if the RES(i) is correct. If so, SGSN successfully authenticates MS. In this way, mutual authentication between MS and SGSN is achieved. In the procedure, CK(i) and IK(i) are generated for protecting the traffic. The sequence number stored in the MS and SGSN guarantees the freshness of CK and IK.

A. Security Weaknesses in UMT AKA

Several security weaknesses in UMTS AKA are discovered, including redirection attack, man-in-the-middle attack, sequence number depletion attack, and sequence number reset attack. With these attacks, the adversary can annoy a victim user with billing problems, or even eavesdrop the communication contents.

With the redirection attack, an adversary can lure a legitimate user to connect to his counterfeit base station by broadcasting with stronger signal a bogus base station ID. At the same time the adversary connects to another legitimate foreign network on behalf of the legitimate user. Unbeknownst to the victim MS, the adversary relays messages between the legitimate foreign network and the victim MS without any modification to the communication contents. Fortunately, the message contents of the victim MS are protected by the cipher key CK and integrity key IK, and therefore the adversary cannot modify them. In this context, the adversary can only redirect the traffic to another network. The victim MS will perform authentication procedure with the foreign network because the foreign network is legitimate.

This redirection attack will persecute a victim MS with billing problems, forcing the victim MS in his home network being charged for roaming into a foreign domain operated by another service provider. In this context, neither can the home network detect that the victim MS is under the redirection attack, nor can the victim MS. Moreover, it is possible that the adversary can redirect the victim MS to a network with weak or no data encryption, such as a false GSM base station. Thus, the adversary can eavesdrop and recover the communication contents **[10]**.

The mounting man-in-the-middle attack is able to lure the victim MS to use a service network with weak encryption or no encryption so that an adversary can eavesdrop the whole communication initiated by the victim MS. The adversary can impersonate a GSM base station and induce the victim MS to establish a connection with him. This kind of attacks can bypass UMTS security mechanism and force GSM/UMTS dual mode cell phone to use GSM authentication procedure, in which the "GSM cipher mode command" message can easily be altered. Unlike the "security mode command" in UMTS authentication procedure, "GSM cipher mode command" in GSM authentication procedure is not protected with integrity key **[11]**.

In addition to the aforementioned attacks, we discovered two more types of replay attacks, namely the sequence number depletion attack and the roaming attack. The Authentication Token AUTN(i) contains a sequence number SQN which can be used by MS to verify the freshness of the token. If the sequence number SQN is in the correct range, the token is accepted. Otherwise, it will be denied. However, two types of attacks may succeed. First, if the sequence number is depleted and started over again, the same sequence numbers will be repeated. In this case, the replay of an old token will succeed. Second, when a MS roams to a SGSN which he visited before, the sequence number may be reset, and the replay of an old token will also succeed. In both cases, the MS cannot verify the freshness of a token.

B. Efficiency Weaknesses of UMTS AKA

In UMTS AKA, after the SGSN sends HLR/AuC the authentication data request, the HLR/AuC replies to the SGSN with *n* authentication vectors AV(i). If the MS stays within the same SGSN long enough until all AVs are exhausted, the SGSN must resend the authentication data request to HLR/AuC for another set of AVs. The transmission of authentication data request and AV consumes a huge amount of bandwidth, and the

authentication data request may be expensive because the SGSN and the HLR/AuC may be located in different countries. Furthermore, the number of AVs sent from the HLR/AuC to the SGSN is also important. For instance, if the MS stays in the same SGSN for a long time, a small *n* will consume much more bandwidth than a larger *n*. However, it is difficult to anticipate the time the MS will stay in the same SGSN, and therefore it is nontrivial to determine an appropriate *n*. The lack of adaptive scheme lowers the efficiency of AKA.

III. PROPOSED SCHEME S-AKA

To cope with the aforementioned problems, a new secure AKA scheme (S-AKA) is proposed. Before elaborating the proposed scheme, we first state the assumptions of the environment, which is consistent with 3GPP **[1]**. The assumptions are: 1) The VLR/SGSN is trusted by the user's home network to handle the authentication information securely, 2) The links between the VLR/SGSN and the HLR/AuC are adequately secure, and 3) the user trusts the HLR/AuC. The goals of our proposed scheme includes the following: 1) defeat the redirection attack, 2) defeat the man-in-the-middle attack, 3) achieve mutual authentication between MS and HLR/AuC, 4) accomplish mutual authentication between MS and SGSN, 5) negotiate a cipher key CK and an integrity key IK, 6) freshness assurance to the user of the established keys, and 7) reduce the bandwidth consumption. With these goals, our proposed scheme has the capability to provide more secure and efficient services. Some symbols and abbreviations used in S-AKA are summarized in Table 1.

Table 1. Symbols and Abbreviations

f6	Key generation function used to compute DK
f7	Key generation function used to compute PLK
AK	Anonymity Key
AMF	Authentication management field
AUTN	Authentication Token
CK	Cipher Key
DK	Delegation Key
FRESH	A random number generated by MS
IK	Integrity Key
K	Long-tem secret key shared between the USIM and the AuC
LAI	Location Area Identity
MAC	The message authentication code generated by f1
PLK	Payload Encryption Key
XRES	Expected Response

S-AKA can resolve the redirection attack with assistance of the MS itself and the SGSN. In S-SKA, the MS can reject illegal base station connection, and on the other hand the SGSN can verify the LAI sent from the MS. If the LAI is illegal, the SGSN will drop the connection. The LAI in UMTS AKA is not encrypted by any means, and thus can be altered by the adversary for the redirection attack. In S-AKA, we use MAC to protect the integrity of LAI. If an adversary attempts to modify LAI, the illegal modification will be detected immediately.

S-AKS can also cope with the man-in-the-middle attack. S-AKA introduces a new key, PLK, to encrypt the payload. Connecting to a GSM BSS, the MS and SGSN generate a PLK to encrypt and decrypt the messages transmitted between them. PLK prevents an adversary to eavesdrop as well as to modify the communication. Since there is no mechanism for generating the PLK in UMTS AKA, we introduce a new key generation function f7 for PLK.

The proposed S-AKA scheme uses a ticket-based authentication scheme for bandwidth reduction **[9][12]**. This ticket-based authentication scheme allows the HLR/AuC to authorize the SGSN for subsequent mutual authentication between SGSN and MS. After the HLR/AuC authenticates the MS for the first time, it sends delegation key DK to SGSN. The SGSN then uses DK for subsequent authentication. The ticket-based authentication scheme benefits from the traffic reduction between the HLR/AuC and SGSN, and thus greatly reduces the number of messages and the bandwidth consumption. Because there is no DK generation function in UMTS AKA, we use a new key generation function f6 to generate DK.

As shown in Figure 1, S-AKA can be divided into two protocols. The first protocol, called S-AKA-I, is the authentication procedure taking place for the first time when the MS and the SGSN authenticate each other. The second protocol, S-AKA-II, is the authentication procedure executed for the sequent authentication between the MS and the SGSN. In the initial authentication using the S-AKA-I protocol, the SGSN will communicate with the HLR/AuC to obtain the authorization and delegation information for the sequent authentication to be used in the S-AKA-II protocol. In the S-AKA-II protocol, the MS and SGSN can authenticate each other without data transmission between SGSN and HLR/AuC, which drastically reduces the bandwidth consumption in the course of the authentication procedure.

The S-AKA-I and S-AKA-II protocols shown in Figure 1 will be explained below.

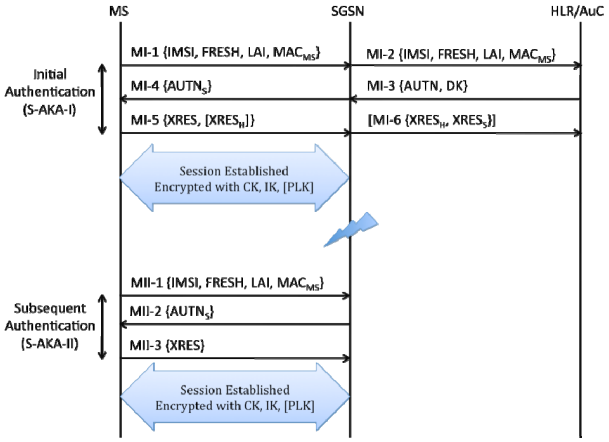

Figure 1. The S-AKA Protocol.

(1) S-AKA-I protocol

Step 1. MS sends IMSI, FRESH, LAI, MAC_{MS} to SGSN

Denoted as MI-1. MS computes DK = $f6_K$(FRESH) with the pre-shared key K. MS sends a registration request to the SGSN through a BSS. The message is comprised of IMSI, FRESH, LAI, and MAC_{MS}. Without loss of generality, IMSI (International Mobile Subscriber Identity) is used herein which is the permanent identifier of a user. By 3G convention, IMSI can be also replaced by the temporary user identifier TMSI (Temporary Mobile Subscriber Identity) to protect user privacy. For simplicity, we will only show the use of IMSI herein. FRESH is a random number generated by MS and will be served as a random challenge for authentication in the protocol. LAI is the location area identifier used to defeat the redirection attack. MAC_{MS} = $f1_K$(FRESH||LAI) is used to protect the integrity of FRESH and LAI.

Step 2. SGSN forwards MI-1 to HLR/AuC

Denoted as MI-2. The SGSN checks if the BSS is physically connected by LAI. If not, the SGSN rejects the request. Otherwise, the SGSN stores the FRESH and forwards IMSI, FRESH, LAI, and MAC_{MS} to HLR/AuC.

Step 3. HLR/AuC sends AUTN and DK to SGSN

Denoted as MI-3. Upon receipt of the request, HLR/AuC generates RAND, DK and computes MAC_H=$f1_K$(RAND||AMF). Then, HLR/AuC generates AUTN=(MAC_H||RAND||AMF) for verifying the legality of the MS. HLR/AuC sends DK and AUTN to the SGSN. In this way, HLR/AuC can successfully delegate SGSN to authenticate the MS for the subsequent authentication in the S-AKA-II protocol.

Step 4. SGSN sends $AUTN_S$ to MS

Denoted as MI-4. SGSN generates $RAND_S$, MAC_S= $f1_{DK}$(MAC_H||$RAND_S$||RAND||FRESH), constructs $AUTN_S$= (MAC_S||$RAND_S$||RAND||AMF||FRESH) and sends it to MS.

Step 5. MS sends $XRES_S$ and $XRES_H$ to SGSN

Denoted as MI-5. MS authenticates the SGSN by verifying MAC_S. The MS checks if the received authenticated response FRESH is equal to his earlier challenge FRESH. To authenticate a response FRESH in $AUTN_S$, $f1_{DK}$(MAC_H||$RAND_S$||RAND||FRESH) is verified. In response to MI-4, MS computes $XMAC_H$=$f1_K$(RAND||AMF) and $XMAC_S$= $f1_{DK}$($XMAC_H$||$RAND_S$||RAND||FRESH) to authenticate both HLR/AuC and SGSN. To authenticate HLR/AuC, the equality of MAC_H and $XMAC_H$ is verified. MS can also authenticate SGSN by checking if $XMAC_S$ is equal to MAC_S. If not, either HLR/AuC or SGSN is invalid, and MS drops the connection. If both are valid, the MS computes $XRES_H$ = f2K(RAND), XRES= $f2_{DK}$($RAND_S$), IK= $f4_{DK}$($RAND_S$), and CK = $f3_{DK}$($RAND_S$). To withstand false GSM BSS attacks, MS checks if a GSM BSS is connected. If so, PLK = $f7_{DK}$($RAND_S$) is used to encrypt payloads before CK and IK to protect the session. Then, the SGSN checks the legitimacy of MS by verifying XRES. The SGSN also computes IK, CK and PLK if it detects a GSM BSS involved in the session.

[Step 6.] SGSN sends $XRES_H$ and $XRES_S$ to HLR/AuC

Denoted as MI-6 (optional). The sixth message is an optional message. SGSN computes $XRES_S$ = $f2_{DK}$(RAND), and sends it back to HLR/AuC together with $XRES_H$ received from MS. Then, HLR/AuC can mutually authenticate the legitimacy of MS and SGSN by verifying $XRES_H$ =$f2_K$(RAND) and $XRES_S$ = $f2_{DK}$(RAND), respectively. Receipt of the two responses from MS and SGSN ensures HLR/AuC that both participants have successfully completed the S-AKA, and acquire all the secrets needed for subsequent authentications. To achieve the mutual authentication only between MS and SGSN, the sixth message is not needed. However, this message serves as a response to HLR/AuC, and provides S-AKA additional security features where HLR/Auc can authenticate the MS and ensure that SGSN receives the security information he sent earlier.

(2) S-AKA-II protocol

S-AKA-II protocol is designed for subsequent authentications. When connecting to the same SGSN for the second time and onwards, S-AKA-II is executed to reduce the bandwidth consumption.

Step 1. MS sends IMSI, FRESH, LAI, MAC_{MS} to SGSN

Denoted as MII-1. The MS utilizes the DK derived in S-AKA-I for subsequent authentications in the same SGSN. The MS generates a random challenge FRESH and sends a request to SGSN through a BSS. This message is similar to MI-1 in S-AKA-I, but uses DK for encryption, instead of K.

Step 2. SGSN sends the $AUTN_S$ to MS

Denoted as MII-2. The parameters (FRESH, $RAND_S$, DK, AMF, MAC_H, and RAND) obtained in S-AKA-I help SGSN and MS authenticate each other without HLR/AuC. 1) The SGSN checks LAI whether the BSS is physically connected. If not, the SGSN rejects the request immediately. 2) The SGSN checks the MAC_{MS} on behalf of the HLR. If the SGSN detects the MAC_{MS} is not valid, the SGSN rejects the connection. Otherwise, the SGSN generates $RAND_S$, MAC_S and $AUTN_S$, and sends the $AUTN_S$ to MS.

Step 3. The MS sends XRES to SGSN

Denoted as MII-3. Similar to MI-5, the MS authenticates the SGSN by verifying MAC_S. Then, MS sends XRES to SGSN. The SGSN verifies the legitimacy of the MS by checking the correctness of XRES.

IV. Security and Efficiency Analysis

Since S-AKA adopted the architecture of UMTS AKA, the security features such as signaling data integrity, user traffic confidentiality, and the ability against various attacks are achieved. Here we only examine additional security features of the proposed S-AKA protocol.

A) Security against redirection attack

We divide the scenario into two cases according to the behavior of the adversary's BSS.

Case 1. Masquerading a BSS in the foreign territory

Assume the adversary's BSS broadcasts the LAI, which is in a foreign territory. Since the MS can monitor the status of the base stations nearby, the MS can choose to connect to those base stations belonging to the home territory. The MS will not connect to the adversary's BSS unless the adversary's BSS jams the whole spectrum to trick the MS to believe that there are no other base stations. However, the user will still discover that he is connecting to a foreign network since the foreign network ID will be shown on the MS.

Case 2. Masquerading a BSS in the home territory

In this case, the MS is not able to distinguish the genuine base station from the adversary's since they are all in the home territory. The adversary's BSS broadcasts its LAI using higher power to entice the MS to connect with him. However, the SGSN or HLR/AuC can help the MS detect this situation. In MI-1, the MS sends LAI to the SGSN or HLR/AuC. Upon receipt of the LAI, the SGSN or HLR/AuC first checks if the base station is indeed physically connected. If not, the request is rejected immediately. Thus, the adversary's BSS, pretending to be in the home network, cannot redirect the connection to a foreign network.

In the above two cases, the redirection attack cannot be carried out in S-AKA. Not only does S-AKA prevent user from suffering billing problems but also avoid being tricked into a network with weak encryption keys.

B) Countermeasure against man-in-the-middle attack

To defeat the man-in-the-middle attack, we introduce an extra key PLK to encrypt payloads. When MS detects that it is connecting to a GSM BSS, it computes the PLK right after receiving MI-4 or M II-2. The MS then encrypts the data using the PLK to provide data confidentiality between the MS and SGSN. Even if the adversary's false GSM BSS chose not to encrypt the data, the PLK can still protect the data confidentiality.

The SGSN also computes the PLK after receiving the MI-5 or MII-3 to decrypt the payload as the SGSN notices the data is received from a GSM BSS. Since the encryption process with PLK involved could be implemented using simple XOR operations, the encrypt/decrypt operations will not consume too much computation power so the efficiency and the data confidentiality will still remain.

C) Mutual authentication between MS and HN

In MI-2, HLR/AuC checks the FRESH and MAC_{MS} to authenticate MS. On the other hand, MS authenticates HN when receiving the $AUTN_S$ from the SGSN (MI-4 or MII-2). By verifying $XMAC_S = f1_{DK}(XMAC_H||RAND_S||RAND||AMF||FRESH)$, MS can authenticate both HN and SGSN.

D) Mutual authentication between MS and SGSN

The SGSN authenticates the MS by verifying the XRES in MI-5 and MII-3. If XRES equals $f2_{DK}(RAND_S)$, the MS is authenticated. When the MS intends to authenticate the SGSN, it first computes the $XMAC_H = f1_K(RAND||AMF)$, and $XMAC_S = f1_{DK}(XMAC_H || RAND_S ||RAND||FRESH)$. The MS then verifies if $XMAC_S$ equals MAC_S. If so, the SGSN is successfully authenticated.

E) Key establishment and freshness assurance

In S-AKA, CK and IK are negotiated in MI-5 and MII-3. FRESH, $RAND_S$ and RAND in the $AUTN_S$ can be used to guarantee the freshness of CK and IK.

F) Security against Replay Attack

Since an adversary can capture MI-1, MI-4, and MI-5 ion S-AKA-1 or MII-1, MII-2, and MII-3 in S-AKA-2, he may attempt to launch a replay attack by replaying these messages. In **MI-1** and **MII-1**, FRESH is defined as a random challenge generated by MS, and is protected by MAC_{MS}. The replayed message will be discovered by MS and the connection will then be dropped. In **M I-4** and **MII-2**, $AUTN_S$ contains MAC_S, $RAND_S$, RAND, AMF, and FRESH. Similarly, if a message is replayed, MS can detect the random challenge FRESH and drop the connection. In **MI-5** and **MII-3,** XRES is derived from $f2_{DK}(RAND_S)$. Since $RAND_S$ changes every authentication, replayed XRES will not be accepted by SGSN.

The use of challenge-response protocol in S-AKA is very effective to detect a replay message. In the context, an authenticated response must match its random challenge. Otherwise, the response is considered a replay. Although it is not a security requirement, some applications may desire to eliminate the replay of a random challenge, generated due to network faults. In this case, we will need to add a sequence number to the S-AKA protocol, where a challenge becomes a 2-tuple vector (random challenge, sequence number). To detect a duplicate random challenge, the 2-tuple vector will be checked against the sequence number stored at the receiver. Upon receipt of the 2-tuple vector, if the sequence number in the challenge vector is not large than the counter stored at the receiver, it is a replay. The early detection of a duplicate challenge can eliminate the computation of an unneeded response.

G) Bandwidth analysis

The lengths of the five UMTS AKA messages (M1 to M5) are listed in Table 2. Two cases listed below may consume different bandwidth.

Table 2. Lengths of the UMTS AKA Messages

Message	Contents	Bits
	Service Request	8
	LAI	40
	IMSI, RAND, CK, IK	128
	XRES	32
	AUTN (of UMTS AKA)	128
M1, M2	IMSI||Service Request||LAI	176
M3	RAND||XRES||CK||IK||AUTN	544 * m
M4	RAND||AUTN	320
M5	RES	32

Case 1. If the SGSN doesn't have any unused AVs, all of the messages must be transmitted. Thus, the bandwidth consumption is

L(M1)+L(M2)+L(M3)+L(M4)+L(M5)
*= 704 + m*544 bits*

Case 2. If the SGSN has unused AVs, only UM1, UM4 and UM5 are transmitted. The bandwidth consumption is

L(M1) + L(M4) + L(M5) = 528 bits

Table 3. Lengths of the S-AKA Messages.

Message	Contents	Bits
	FRESH	24
	MAC_{MS}	64
	AUTN (of S-AKA)	208
MI-1, MI-2, MII-1	IMSI||Service Request||LAI||FRESH|| MAC_{MS}	264
MI-3	AUTN||DK	336
MI-4, MII-2	RAND||AUTN	360
MI-5	XRES|| $XRES_H$	64
[MI-6]	$XRES_H$|| $XRES_S$	64
MII-3	XRES	32

Table 3 lists the bandwidth consumptions of S-AKA. Similarly, there are two cases, which may consume different bandwidth.

Case 1. If it is the first time the MS meets the SGSN, the S-AKA-I must be performed. The bandwidth consumption is
L(MI-1)+L(MI-2)+L(MI-3)+L(MI-4)+L(MI-5)+L(MI-6)
=1288 bits
Since MI-6 is optional, the overhead can be further reduced. If MI-6 is skipped, the bandwidth consumption becomes 1256 bits.

Case 2. If it is not the first time MS wants to authenticate with the SGSN, the S-AKA-II will be executed and the bandwidth consumption is

L(MII-1)+L(MII-2)+L(MII-3) = 656 bits.

From the analysis, we conclude that S-AKA improves the communication efficiency by reducing 40% or even 45% (if MI-6 is skipped) of the bandwidth consumption. This is a significant improvement as a large number of MS use the authentication services.

H) Scalability analysis

In UMTS AKA, CK and IK are used to protect the communication session between MS and SGSN. When MS is authenticated, it must perform *n* encryptions and integrity checks for a session with *n* messages. Compared to UMTS AKA, an extra encryption key PLK is used only if GSM BSS is involved in a communication session. In this case, MS in S-AKA needs *n* extra encryptions for the session with *n* messages. However, we can reduce such a burden by adopting the Exclusive-OR operator on the implementation of the encryption process with PLK.

V. SECURITY PROOF OF S-AKA

To prove the security of our scheme, we formalize our model in a similar fashion to Muxiang's security model [4] and Shoup's formal security model [13]. We first define some preliminaries, elaborate the security model, and finally prove the security of S-AKA.

A. Preliminaries

Let $\{0,1\}^n$ denote the set of binary strings of length n and $\{0,1\}^{\leq n}$ denote the set of binary strings of length at most n. For two binary strings $s1$ and $s2$, the concatenation of $s1$ and $s2$ is denoted by $s1||s2$. A real-valued function $\varepsilon(k)$ of non-negative integers is called *negligible* (in k) if for every $c > 0$, there exists $k_0 > 0$ such that $\varepsilon(k) \leq 1/k_c$ for all $k > k_0$. Let $X = \{X_k\}_{k\geq 0}$ and $Y = \{Y_k\}_{k\geq 0}$ be sequences of random variables, where X_k and Y_k take values in a finite set S_k. For a probabilistic polynomial time algorithm D that outputs 0 or 1, we define the *distinguishing advantage* of D as

$$Adv^{dist}_{X_k,Y_k}(D) = \left|\Pr(D(X_k)=1) - \Pr(D(Y_k)=1)\right|$$

If for every probabilistic polynomial-time algorithm, the distinguishing advantage is negligible in k, we say that X and Y are *computationally indistinguishable*.

Let $G : \{0,1\}^k \times \{0,1\}^d \rightarrow \{0,1\}^s$ denote a family of functions and let $U(d, s)$ denote the family of all functions from $\{0,1\}^d$ to $\{0,1\}^s$. For a probabilistic polynomial-time oracle machine A, the *prf-advantage* of A is defined as

$$Adv^{prf}_G(A) = \left|\Pr(g \xleftarrow{R} G : A^g = 1) - \Pr(g \xleftarrow{R} U(d,s) : A^g = 1)\right|$$

,

where $g \xleftarrow{R} G$ denotes the operation of randomly selecting a function g from the family G. We associate to G an insecurity function:

$$Adv^{prf}_G(t,q) = \max_{A \in A(t,q)} Adv^{prf}_G(A),$$

where $A(t, q)$ denotes the set of adversaries that make at most q oracle queries and have running time at most t. Assume that d and s are polynomials in k. If for every probabilistic polynomial-time oracle machine A, $Adv^{prf}_G(A)$ is negligible in k, then we say that G is a *pseudorandom* function family.

A *Message Authentication Code* is a family of functions F and $f1 \in$ F of $\{0,1\}^k \times Dom(f1)$ to $\{0,1\}^l$, where $Dom(f1)$ denotes the domain of $f1$. In this paper, $Dom(f1) = \{0,1\}^{\leq L}$. For $K \in \{0,1\}^k$ and $M \in \{0,1\}^{\leq L}$, let $\sigma = f1(K,M)$. We refer to σ as the MAC of M. For the security of $f1$, we use the notion of security against chosen message attacks. An adversary is a probabilistic polynomial-time algorithm which has access to an oracle that computes MAC under a randomly chosen key K. We define the *mac-advantage* of an adversary A, denoted by $Adv^{mac}_F(A)$, as the probability that $A^{f1(K,)}$ outputs a pair (σ,M) such that $\sigma = f1(K,M)$, and M was not a query of A to its oracle. We associate to F an insecurity function,

$$Adv^{mac}_F(t,q) = \max_{A \in A(t,q)} Adv^{mac}_F(A)$$

where $A(t, q)$ denotes the set of adversaries that make at most q oracle queries and have running time at most t. If for every polynomially bounded adversary A, $Adv^{mac}_F(A)$ is negligible in k, we say that $f1$ is a secure message authentication code.

B. Security Model

The security model consists of two systems, an ideal system and a real system. Security is based on simulatability of adversaries in the two systems. The ideal system follows Shoup's formal model of security for authenticated key exchange in the two-party setting, and Muxiang's security model [4]. The real system is adapted from Shoup's formal model of security for authenticated key exchange in the three-party setting.

C. Security proofs

Following are four definitions. With these definitions, we can make the proof more concise and understandable.

Definition 1. Let I_{ij} be an entity instance in the real system. A *stimulus* on I_{ij} is a message such that the status of I_{ij} changes from continue to accept after receiving the message.

Definition 2. Let A be a real world adversary and let T_A be the transcript of A. For every accepted instance I_{ij}, if the stimulus on I_{ij} was output by a compatible instance, we say that T_A is an *authentic transcript*.

Definition 3. Let A be a real-world adversary and let T_A be the transcript of A. In the game of A, if the random numbers generated by an entity and its instances are different, we say that T_A is a *collision-free transcript*.

Let |RAND| and $|RAND_S|$ denote the length of RAND and $RAND_S$, respectively. Assume that these numbers are randomly selected in the game of A. Let C_A denote the event that T_A is collision-free. Then

$$\Pr(\overline{C_A}) \leq \frac{n_i^2(2^{-|RAND|} + 2^{-|RAND_S|})}{2} \qquad (5.1)$$

where n_i denotes the number of instances initialized by A. In the following, we assume that |RAND| and $|RAND_S|$ are polynomials in k, then $\Pr(\overline{C_A})$ is negligible.

Definition 4. Let T_A be the transcript of a real-world adversary A. Let $\sigma_1, \sigma_2, \ldots\sigma_n$ denote all the tags which are computed under f1 by entities and entity instances. If $\sigma_i \neq \sigma_j$ for any $i \neq j$, we say that f1 is collision-resistant in T_A.

Lemma 1. *Let A be a real-world adversary and let T_A be the transcript of A. Assume that T_A is collision-free. Also*

assume that f1 and f2 are independent function families and are collision-resistant in T_A. Let M_A denote the event that T_A is authentic. Then

$$\Pr(\overline{M_A}) \le n_i(2 * Adv_F^{mac}(t,q))$$

Proof. If T_A is not authentic, there exists at least one instance, which has accepted, but the stimulus on this instance was not output by a compatible instance. We claim that the probability of such an event is upper-bounded by $\Pr(\overline{M_A}) \le n_i(2 * Adv_F^{mac}(t,q))$. To prove our claim, we consider the following three cases.

Case 1. Let $I_{i'j'}$ be the network instance which has received the message (IMSI, FRESH, LAI, MAC_{MS}) and has accepted. Since the identity $IMSI_{i''}$ is used in the computation of the MAC_{MS}, the stimulus on $I_{i'j'}$ could not be output by a user instance not compatible with $I_{i'j'}$. We can then construct an adversary A_F for the message authentication code F. The adversary A_F has oracle access to $f1_K$ and $f2_K$, where K was chosen at random. Assume that $IMSI_{i'j'}$ is assigned to a user U, which may or may not be initialized by A. The adversary A_F begins its experiment by selecting authentication keys for all users, except that the authentication key for user U is not chosen. Next, A_F runs A just as in the real system. In the game of A, if an entity or entity instance needs to evaluate f1 and f2 under the key of U, A_F provides the evaluation by appealing to the oracles $f1_K$ and $f2_K$. If an entity or entity instance needs to evaluate f3, f4, f6, f7 under the key of U, A_F supplies a random number or even a constant for the evaluation. If at any point $I_{i'j'}$ accepts, A_F stops and outputs (MAC_{MS}, FRESH||LAI), else A_F stops at the end of the game of A and output an empty string.

Let $Succ(A_F, F)$ denote the event that A_F outputs a MAC and a message and the message was not queried to the oracle $f1_K$. Let $AS_{i'j'}$ denote the event that $I_{i'j'}$ has accepted, but the stimulus on $I_{i'j'}$ was not output by a user instance. If $AS_{i'j'} = 1$, the A_F has successfully forged the MAC for the message FRESH||LAI and this message was not queried to the oracle $f1_k$. This implies that

$$\Pr(AS_{i'j'} = 1) \le \Pr(Succ(A_F, F)) = 1 \qquad (5.2)$$

$$\Pr(AS_{i'j'} = 1) \le Adv_F^{mac}(t,q) \qquad (5.3),$$

where t=O(T), q=O(n_i).

Case 2. Let I_{ij} be a user instance which has received the message (AUTNs) and has accepted. Let AS_{ij} denote the event that the stimulus on I_{ij} was not output by a network instance. Let IS_{ij} denote the event that the stimulus on I_{ij} was output by a network instance $I_{p'q'}$ but not compatible with I_{ij}. If IS_{ij} is true, then the instance $I_{p'q'}$ received the message (IMSI, FRESH, LAI, MAC_{MS}) before sending out $AUTN_S$, where $AUTN_S = MAC_S||RAND_S||RAND||AMF||FRESH$, and $MAC_S = f1_{DK}(MAC_H||RAND||AMF)$. Since T_A is collision-free, $RAND_S$ and RAND can not be generated by a user instance other than I_{ij}. This implies that the adversary A has successfully concocted the MAC_{MS}. By (5.3), we have

$$\Pr(IS_{ij} = 1) \le Adv_F^{mac}(t,q), \qquad (5.4)$$

where t=O(T), q=O(n_i).

Now suppose that AS_{ij} is true, then the adversary A has successfully concocted the MAC_H and MAC_S. Running the adversary A, we can construct an adversary A'_F for f1. The adversary A'_F works in the same way as f1 except that, when I_{ij} accepts, A'_F stops and outputs two pairs: (MAC_H,RAND||AMF), and (MAC_S, MAC_H||RANDs|| FRESH||RAND). Using the notation *Succ(A'F, F)* as described above, we have

$$\Pr(AS_{ij} = 1) \le \Pr(Succ(A'_F, F) = 1) \qquad (5.5)$$

Therefore, by (5.4) and (5.5), the probability that the stimulus on a user instance I_{ij} was not output by a compatible network instance is upper-bounded by

$$\Pr(AS_{ij} = 1) + \Pr(IS_{ij=1}) \le 2 * Adv_F^{mac}(t,q) \qquad (5.6)$$

Case 3. Let $I_{i''j''}$ be a network instance which has received (XRES) and has accepted, where $RAND_S$ was sent out by $I_{i''j''}$ in the $AUTN_S$. If the stimulus on $I_{i''j''}$ was not output by a user instance, then the adversary A has successfully concocted the XRES. Similar to (5.3), it can be proved that the probability of such an event is upper-bounded by $Adv_F^{mac}(t,q)$. Next, if the stimulus on $I_{i''j''}$ was output by a user instance I_{pq} which is not compatible with $I_{i''j''}$. Then the user instance I_{pq} received $AUTN_S$ before it output the stimulus. Since T_A is collision-free, $AUTN_S$ cannot be output by a network instance other than $I_{i''j''}$. This means that it is the adversary who concocted the MAC_S for $MAC_H||RAND_S||FRESH||RAND$. By (5.5), the probability of such an event is upper-bounded by $2 * Adv_F^{mac}(t,q)$.

Based on the above analysis, it can be concluded that the probability that T_A is not an authentic transcript is at most $n_i(2 * Adv_F^{mac}(t,q))$, where n_i is the number of instances. □

Lemma 2. *Let A be a real-world adversary and let T_A be the transcript of A. Assume that T_A is authentic and collision-free. Also assume that G is a pseudorandom function family, independent of f1, and f1 is collision-resistant in T_A. Then there exists an ideal-world adversary A* such that for every distinguisher D with running time T,*

$$Adv_{T_A, T_A^*}^{dist}(D) \le n_e \; Adv_G^{prf}(t,q)$$

where n_e is the number of user entities initialized by A and n_i is the number of instances initialized by A, t=O(T), q=O(n_i)

Proof. We construct a simulator that takes the real-world adversary A as input and creates an ideal-world adversary A*. The simulator basically has A* run the adversary A just as in the real system. For any implementation record in the real-world transcript, A* copies this record into the ideal-world transcript by issuing an implementation operation. Corresponding to each (start session, i,j) record that A's action cause to be placed in the real-world transcript, A* computes a connection assignment, and the ring master in the ideal system substitutes the session key SK_{ij} by an idealized session key K_{ij}, which is a random number. Corresponding to each (abort session, i,j) record that A's action cause to be placed in the real-world transcript, A* executes the operation (abort session, i,j). For an application operation, the ringmaster in the ideal system makes the evaluation using the idealized session keys. This way, we have an ideal-world adversary whose transcript is almost identical to the transcript of the real-world adversary A. The differences exist in the application records. In the following, we show that the connection assignments made by A* are legal and the differences between the two transcripts are computationally indistinguishable.

Case 1. Assume that a user instance I_{i1j1} has received the message ($AUTN_S$) and has accepted, where $AUTN_S$ = $MAC_S||RAND_S||RAND||AMF||FRESH$. Since T_A is authentic, this message must be output by a network instance $I_{i1'j1'}$ compatible with I_{i1j1}. In this case, we let the adversary A* make the connection assignment (create, i_1',j_1'). We have to argue that this connection assignment was not made before. This is true because $AUTN_S$ could not be a stimulus on other user instances, otherwise the MAC_S would not be acceptable by I_{i1j1}. So it is legal for the adversary A* to make the connection assignment. Consequently, it is also legal to substitute the session key SK_{i1j1} by a random number K_{i1j1}.

Case 2. Assume that a network instance $I_{i2'j2'}$ has received the message (IMSI, FRESH, LAI, MAC_{MS}) from a user instance I_{i2j2} and has accepted, where MAC_{MS} = $f1_{k_{i2}}$(FRESH|| LAI). In this case, we let A* makes the connection assignment (create, i_2,j_2) and let the ring master substitute the session key $SK_{i2'j2'}$ by a random number $K_{i2'j2'}$. Since f1 is collision-resistant in T_A, MAC_{MS} could not be a stimulus on any instances other than $I_{i2'j2'}$. So the connection assignment (create, i_2, j_2) was not made before.

Case 3. Assume that a network instance $I_{i3'j3'}$ has received the message (XRES) from a user instance I_{i3j3} and has accepted, where XRES = $f2_{K_{i3}}(RAND_S)$, $RAND_S$ was sent out by $I_{i3'j3'}$. Under the assumption that T_A is collision-free and f2 is collision-resistant in T_A, it can be concluded that I_{i3j3} has accepted and the stimulus on I_{i3j3} was output by $I_{i3'j3'}$. According to Case 1, I_{i3j3} has been isolated for $I_{i3'j3'}$. So it is legal for A* to make the connection assignment (connect, i_3,j_3). Accordingly, the ringmaster sets $K_{i3'j3'}$ by K_{i3j3}.

The above analyses show that there exists a connection assignment for each start session record in T_{A*}. Next, we show that the two transcripts T_A and T_{A*} are computationally indistinguishable. Note that if we remove the application records in both T_A and T_{A*}, then the remaining transcripts are exactly the same. So we only need to consider the application records in both transcripts. First, let's assume that there is only one user entity initialized by A. Let D be a distinguisher for T_A and T_{A*}. By running D on T_A and T_{A*}, we have an adversary D' for G (including f3, f4, f7) such that

$$Adv^{dist}_{T_A,T^*_A}(D) = Adv^{prf}_G(D') \le Adv^{prf}_G(t,q)$$

where t = O(T), q=O($2n_i$), n_i is the number of instances initialized by A.

Now, assume that the number of user entities initialized by A in n_e. Let K_1, K_2, …, K_{ne} denote the keys of there user entities. Then D and D' have access to the input-and-output pairs of G_{K1}, G_{K2}, …, G_{Ke}. It can be concluded that

$$Adv^{dist}_{T_A,T^*_A}(D) \le n_e Adv^{prf}_G(t,q),$$

which proves the lemma. □

Theorem 1. *Assume that G is a pseudorandom function family, f1 is a secure message authentication code, and G, and f1 are independent. Then S-AKA is a secure authentication and key agreement protocol.*

Proof. The completion requirement follows directly by inspection. Now we prove that the simulatability requirement is also satisfied. Let A be a real world adversary and let T_A be the transcript of A. Since f1 is a secure message authentication code, the probability that f1 is not collision-resistant is negligible. Without loss of generality, let's assume that f1 is collision-resistant in T_A. By Lemma 2, there exists an ideal world adversary A* such that for every distinguisher D with running time T,

$$\left|\Pr(D(T_A)=1 \mid M_A \cap C_A) - \Pr(D(T_{A^*})=1 \mid M_A \cap C_A)\right| \le n_e Adv^{prf}_G(t,q)$$

Thus, it follows that

$$Adv^{dist}_{T_A,T^*_A}(D) = \left|\Pr(D(T_A)=1) - \Pr(D(T_{A*})=1)\right|$$
$$\le n_e Adv^{prf}_G(t,q) + \Pr(\overline{M_A}) + \Pr(\overline{C_A})$$

Therefore,

$$Adv^{dist}_{T_A,T^*_A}(D) \le n_e Adv^{prf}_G(t,q) + \Pr(\overline{M_A} \mid C_A) + 2\Pr(\overline{C_A})$$

By (5.1), $\Pr(\overline{C_A})$ is negligible in *k*. By Lemma 1, $\Pr(\overline{M_A} \mid C_A)$ is also negligible. Hence, $Adv^{dist}_{T_A,T^*_A}(D)$ is

negligible. S-AKA is a secure authentication and key agreement protocol. □

VI. CONCLUSION

In this paper, we first introduce the four security weaknesses of UMTS AKA, namely, vulnerabilities to redirection attack, man-in-the-middle attack, sequence number depletion attack, and roaming attack, along with the bandwidth bottleneck of UMTS AKA. To cope with the problems, we propose a new secure authentication key agreement scheme, S-AKA, which is more efficient and can defeat the four attacks. We also analyze the security and bandwidth consumption of S-AKA, and compare it with UMTS AKA. The analysis shows that our proposed S-AKA not only defeats those four attacks mentioned above, but also reduces up to 45% of bandwidth consumption. To ensure the security strength of the proposed scheme, we formally prove that S-AKA is a secure authentication and key exchange protocol.

ACKNOWLEDGMENT

This work was supported in part by National Science Council under grant NSC 95-2218-E-001-001, Taiwan Information Security Center (TWISC) under grant NSC 95-2218-E-011-015, iCAST under grant NSC97-2745-P-001-001, Industrial Technology Research Institute (ITRI), Chung Shan Institute of Science and Technology, Ministry of Economic Affair, and Institute for Information Industry.

REFERENCES

[1] 3GPP. 3rd generation partnership project; Technical specification group services and system aspects; 3g security; Security Architecture. Tech. Spec. 3G TS 33.102 V3.7.0, 2000.

[2] European Telecommunications Standards Institute (ETSI). GSM 02.09: Security Aspects, 19939.

[3] 3GPP. 3rd generation partnership project; Technical specification group SA WG3; A Guide to 3rd Generation Security. Tech. Spec. 3G TR 33.900 V1.2.0, 2000.

[4] Muxiang Zhang. Provably-Secure Enhancement on 3GPP Authentication and Key Agreement Protocol.

[5] Győző Gódor, and Sándor Imre Dr. Novel Authentication Algorithm – Public Key Based Cryptography in Mobile Phone Systems. IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.2B, February 2006

[6] Ja'afer Al-Saraireh, and Sufian Yousef. A New Authentication Protocol for UMTS Mobile Networks. EURASIP Journal on Wireless Communications and Networking, VOL 2006

[7] Dong Chun Lee, Hyo Young Shin, Joung Chul, and Jae Young Koh. Improved Authentication Scheme in W-CDMA Networks. ICCSA 2005

[8] Chun-I, Pei-Hsiu Ho, and Hsin-Yu Chen. Nested One-Time Secret Mechanisms for Fast Mutual Authentication in Mobile Communications, IEEE Wireless Communications and Networking Conference, 2007.

[9] Chung-Ming Huang, and Jian-Wei Li. Authentication and Key Agreement Protocol for UMTS with Low Bandwidth Consumption. Proceedings of the 19th International Conference on Advanced Information Networking and Applications(AINA'05)

[10] Muxiang Zhang, and Yuguang Fang. Security Analysis and Enhancements of 3GPP Authentication and Key Agreement Protocol. IEEE Transactions on Wireless Communications, VOL.4, 2005

[11] Ulrike Meyer, Susanne Wetzel. A Man-in-the-Middle Attack on UMTS. WiSe'04, October 1, 2004

[12] C. -C. Lee, M. -S. Hwang, and W. -P. Yang. Extension of authentication protocol for GSM. Communications, IEE Proceedings, April 2003

[13] Victor Shoup. On Formal Models for Secure Key Exchange. 1999

[14] C.T. Lin, S.P. Shieh, "Chain Authentication in Mobile Communication Systems," Vol. 13, Journal of Telecommunication Systems, 2000.