S-AKA: A Provable and Secure Authentication Key Agreement Protocol for UMTS Networks

Yu-Lun Huang, *Member, IEEE*, Chih-Ya Shen, and Shiuhpyng Winston Shieh, *Senior Member, IEEE*

***Abstract*—The authentication and key agreement (AKA) protocol of Universal Mobile Telecommunication System (UMTS), which is proposed to solve the vulnerabilities found in Global System for Mobile Communications (GSM) systems, is still vulnerable to redirection and man-in-the-middle attacks. An adversary can mount these attacks to eavesdrop or mischarge the subscribers in the system. In this paper, we propose a secure AKA (S-AKA) protocol to cope with these problems. The S-AKA protocol can reduce bandwidth consumption and the number of messages required in authenticating mobile subscribers. We also give the formal proof of the S-AKA protocol to guarantee its robustness.**

***Index Terms*—Authentication protocol, key agreement protocol, Universal Mobile Telecommunication System (UMTS) networks.**

I. Introduction

With the boost of mobile applications, third-generation (3G) technology has been widely deployed to modern mobile devices as an improvement to service capabilities, worldwide operations, and performance. As one of the 3G technologies, the Universal Mobile Telecommunication System (UMTS), which is an evolution of the Global System for Mobile Communications (GSM), uses the same core network standard as GSM. Meanwhile, UMTS has been also developed into a fourth-generation (4G) technology. For backward compatibility, these mobile devices also support second-generation (GSM) technology.

To improve the security weaknesses in GSM [1], UMTS authentication and key agreement (AKA) was proposed at the network level [2] for authenticating 3G mobile subscribers. UMTS AKA negotiates security keys between a subscriber and the serving network and then achieves mutual authentication between the two parties. UMTS AKA can successfully defeat most of the vulnerabilities found in GSM systems and provide a more secure telecommunication system. Nevertheless, it is still vulnerable to some attacks, such as redirection [3] and man-in-the-middle attacks [4]. Mobile subscribers may be mischarged or eavesdropped under these attacks.

In this paper, we propose a new AKA protocol that eliminates vulnerabilities and enhances bandwidth efficiency. We also give the formal proof of our protocol to show its security strength. This paper is organized as follows. In Section II, we introduce UMTS AKA and analyze its security and bandwidth bottlenecks. Section III gives the related work and our motivation. In Section IV, we present the secure AKA (S-AKA) protocol. The security analysis and formal proof are given in Sections V and VI, respectively. Section VII concludes this paper.

II. Universal Mobile Telecommunication System Authentication and Key Agreement

The three major entities involved in UMTS AKA [5] are the mobile station (MS), the Serving GPRS Support Node (SGSN), and the Home Location Register/Authentication Center (HLR/AuC). The MS acts on behalf of a mobile subscriber to communicate with the SGSN and HLR/AuC for mutual authentication. The SGSN represents the visited serving network, and the HLR/AuC in the home domain is in charge of subscriber authentication. In UMTS AKA, the MS and HLR/AuC share a secret key K and maintain sequence numbers $\mathrm{SQN}_{\mathrm{MS}}$ and $\mathrm{SQN}_{\mathrm{HN}}$ for resisting replay attacks. The MS and HLR/AuC also execute some cryptographic functions for key generations and integrity checks. Tables I and II define the abbreviations and cryptographic functions used in this paper, respectively.

Five messages are exchanged during authentication in UMTS AKA [5].

UM_1: The MS sends a registration request containing its IMSI to the SGSN via a base station subsystem (BSS). The BSS then handles traffic and signaling between the MS and the GSM core network.

UM_2: The SGSN forwards the request to the HLR/AuC.

UM_3: After authenticating the MS, the HLR/AuC sends an ordered array of m AVs to the SGSN. Each AV consists of RAND, XRES, CK, IK, and AUTN.

UM_4: The SGSN selects an unused AV, retrieves RAND and AUTN, and sends them to the MS.

UM_5: After successfully checking the freshness and correctness of SQN and MAC in AUTN, the MS authenticates the networks and generates RES, CK, and IK for mutual authentication and session protection.

Manuscript received November 17, 2010; revised April 22, 2011 and June 2, 2011; accepted August 7, 2011. Date of publication September 22, 2011; date of current version December 9, 2011. This work was supported in part by the Industrial Technology Research Institute; by the Institute for Information Industry; by Chunghwa Telecom; by the Chung-Shan Institute of Science and Technology; by the Networked Communications Program Office; by the International Collaboration for Advancing Security Technology Program, National Science Conncil of Taiwan, under Grant NSC97-2745-P-001-001; and by the Taiwan Information Security Center, National Science Council of Taiwan, under Grant NSC99-2218-E-009-017 and Grant NSC100-2219-E-009-005. The review of this paper was coordinated by Dr. L. Chen.

Y.-L. Huang is with the Department of Electrical Engineering, National Chiao Tung University, Hsinchu 300, Taiwan.

C.-Y. Shen was with the Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu 300, Taiwan. He is now with the Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan.

S. W. Shieh is with the Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu 300, Taiwan.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2011.2168247

TABLE I
SYMBOLS AND ABBREVIATIONS

Symbol	Definition	bits
ACC	Accumulator	24
AMF	Authentication Management Field	48
AUTN	Authentication Token	Variable
IMSI	International Mobile Subscriber Identity	128
LAI	Location Area Identity	40
MAC	Message Authentication Code	64
RAND	Random Challenge	128
RES	Response	64
XRES	Expected Response	64
XMAC	Expected Message Authentication Code	64
AK	Anonymity Key	48
CK	Cipher Key	128
DK	Delegation Key	128
IK	Integrity Key	128
SK	Secret Key, shared by MS and HLR/AuC	128
PLK	Payload Encryption Key	128

- Since *AUTN* is defined differently in each protocol, its length varies. For example, the length of AUTN in UMTS AKA is 160 bits, 240 bits in Cocktail-AKA and S-AKA, 224 bits in AP-AKA, etc.

TABLE II
CRYPTOGRAPHIC FUNCTIONS

Functions	Definition
$f1$	Message authentication function for *MAC*
$f2$	Message authentication function for *RES* and *XRES*
$f3$	Key generating function for *CK*
$f4$	Key generating function for *IK*
$f5$	Key generating function for *AK*
$f6$	Key generation function for *DK*
$f7$	Key generation function for *PLK*
$\lvert X \rvert$	Length of entity *X*
$\Vert$	Concatenation

A. Security Vulnerabilities

Recently, UMTS AKA has been found vulnerable to redirection and man-in-the-middle attacks [6]. When a mobile subscriber is under attack, an adversary can eavesdrop the communication between the MS and the SGSN or even annoy the MS with billing problems.

1) Redirection Attack: Redirection attack is one of the possible attacks on multihomed mobile networks. In this attack, an adversary owns a device that can simultaneously impersonate both the BSS and the MS [3] at the same time. To deceive the victim MS, the adversary masquerades as a legitimate BSS by broadcasting a bogus BSS ID. It also disguises as the victim MS to trick the BSS (see Fig. 1). The adversary connects to another legitimate foreign network on behalf of the legitimate MS and builds up a transparent tunnel to relay messages between the legitimate foreign network and the victim MS. Since AUTN, RAND, and secret keys are successfully negotiated, the victim MS will then be authenticated by the foreign network.

The redirection attack persecutes a victim MS with billing problems, forcing the victim MS on his home network to be charged for roaming into a foreign domain operated by another service provider. In this case, neither the home network nor the victim MS can detect the redirection attack. It is also possible that the adversary can redirect the victim MS to an insecure network with weak or none encryption. Hence, the adversary can eavesdrop the communication sessions [6].

2) Man-in-the-Middle Attack: In this attack, the adversary lures the victim MS to use a serving network with weak or none encryption. Upon attacking the network, an attacker, which is hiding between the MS and the SGSN, tries to bypass the UMTS security, forces a UMTS/GSM dual-mode mobile device to use the less secure GSM authentication, and obtains AUTN. The attacker can then eavesdrop the session initiated by the victim MS [4]. This makes the attacker easily alter and eavesdrop the unprotected messages and sessions.

B. Bandwidth Consumption

In UMTS AKA, the HLR/AuC sends m AVs to the SGSN after authenticating the MS. The SGSN needs to request for another authentication when these AVs are exhausted. Transmitting authentication requests and AVs, however, requires a high bandwidth and incurs a high communication cost, particularly when the SGSN and HLR/AuC are located in different administrative domains. Obviously, the number of AVs m sent by the HLR/AuC has a great impact on bandwidth consumption. Smaller m not only means less bandwidth consumption for each registration but implies more frequent authentication requests and vector transmission as well. The tradeoff exists due to the difficulties in choosing an optimal m value for the entire network. Fig. 2 shows the bandwidth consumption of UMTS AKA with different values of m.

Taking “50 authentication requests initiated by one MS” as an example, if $m = 2$, it needs 57 472 bits for authentication, but it needs additional 22 976 bits if $m = 100$. For 200 authentication requests, it needs 231 472 bits for each MS (if $m = 2$) but only 214 224 bits if $m = 100$. If millions of MSs are requesting authentications at the same time, the mobile network must offer adequate bandwidth to accommodate these requests.

III. RELATED WORK

Many AKA protocols [7]–[11] were proposed to ensure the authenticity of communication parties and protect mobile communications at different levels, namely, application, device, and network levels. Some of them [12] and TBAS [2] protect transactions at the application level, some schemes [13] discuss device-based authentication that works by registering a device before it can access any service, whereas some others [5] intend to authorize the MS to use a UMTS network resource at the network level. However, when trying to adapt these protocols to UMTS networks, they either do not address the characteristics of UMTS networks or inefficiently perform on authenticating a mobile user in the registration procedure.

Since asymmetric key cryptography requires higher costs in installation and deployment, many symmetric key-based protocols [6], [14]–[17] were proposed to enhance the security of UMTS AKA and to reduce the bandwidth consumption of authentication. In the aforementioned protocols, a secret key is generally shared between the MS and the HLR/AuC. In 2005, X-AKA [14], which is a symmetric key-based authentication protocol, was proposed to prune off the transmission of AVs in UMTS AKA and improves its bandwidth consumption. However, it does not resist redirection and man-in-the-middle

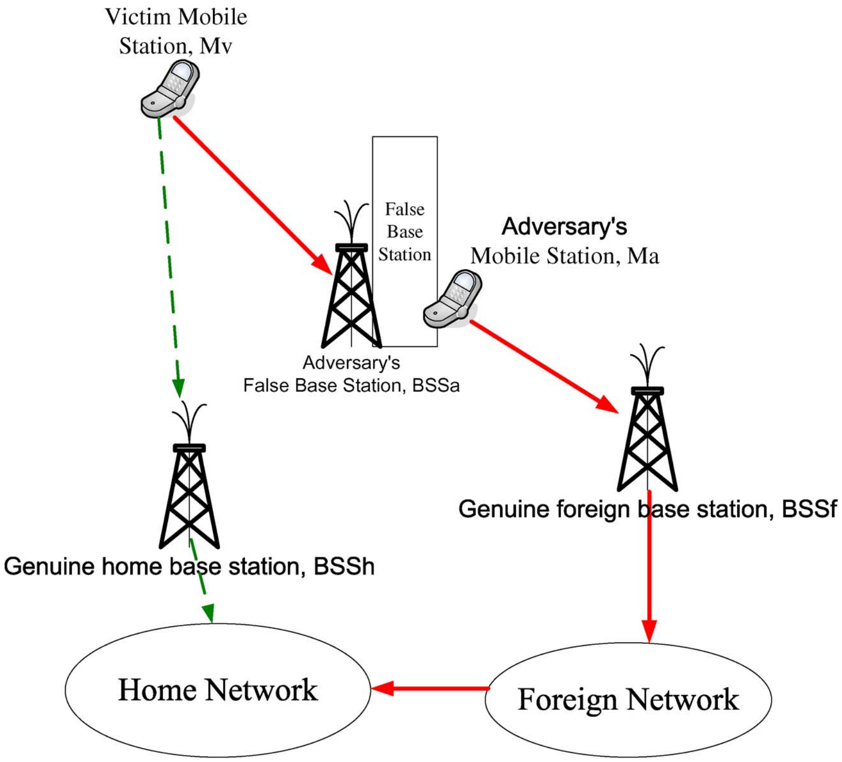
Victim Mobile Station, Mv
False Base Station
Adversary's Mobile Station, Ma
Adversary's False Base Station, BSSa
Genuine foreign base station, BSSf
Genuine home base station, BSSh
Home Network
Foreign Network

Fig. 1. Redirection attack in UMTS AKA.

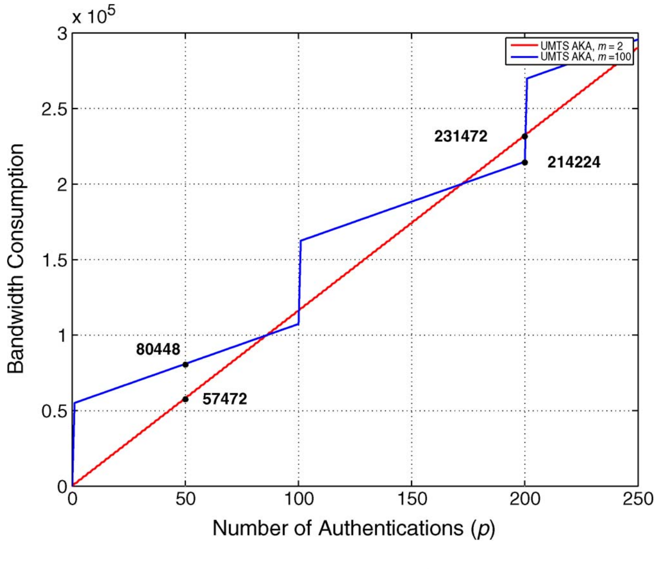

Fig. 2. Bandwidth consumption of UMTS AKA for different numbers of AVs $(m = 2\,100)$.

attacks. Zhang and Fang [6] presented the AP-AKA protocol to defeat the redirection attack and drastically lower the impact of network corruption, but an extra message is initiated by the SGSN for authenticating the roaming MSs. Such a design helps AP-AKA defeat the redirection attack. The protocol, however, is still susceptible to the man-in-the-middle attack.

Al-Saraireh and Yousef [15] proposed a symmetric key-based authentication protocol for UMTS networks. Al-Saraireh and Yousef's protocol mainly focuses on reducing the bandwidth required for transmitting AVs. Hence, the AVs are generated by MSs instead of by serving networks. Al-Saraireh and Yousef's protocol eliminates the cost of delivering AVs during authentication. The protocol, however, does not resolve the security issues in defeating redirection and man-in-the-middle attacks.

In 2010, Ou *et al.* [17] proposed Cocktail-AKA to overcome the congenital defects of UMTS AKA. Cocktail-AKA uses two varieties of AVs (called MAV and PAV) to produce several effective AVs. In the 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. However, Cocktail-AKA is vulnerable to denial-of-service (DoS) attacks [18].

IV. Proposed Protocol: Secure–Authentication and Key Agreement (S-AKA)

We propose an S-AKA, trying to

- defeat redirection and man-in-the-middle attacks;
- mutually authenticate the MS and HLR/AuC, as well as the MS and SGSN;
- negotiate a cipher key CK and an integrity key IK;
- assure the freshness of user keys;
- reduce the bandwidth required for authentication.

S-AKA retains the framework of UMTS AKA with three assumptions.

1) The SGSN can handle user authentication securely.
2) The communication link between the SGSN and the HLR/AuC is secure.

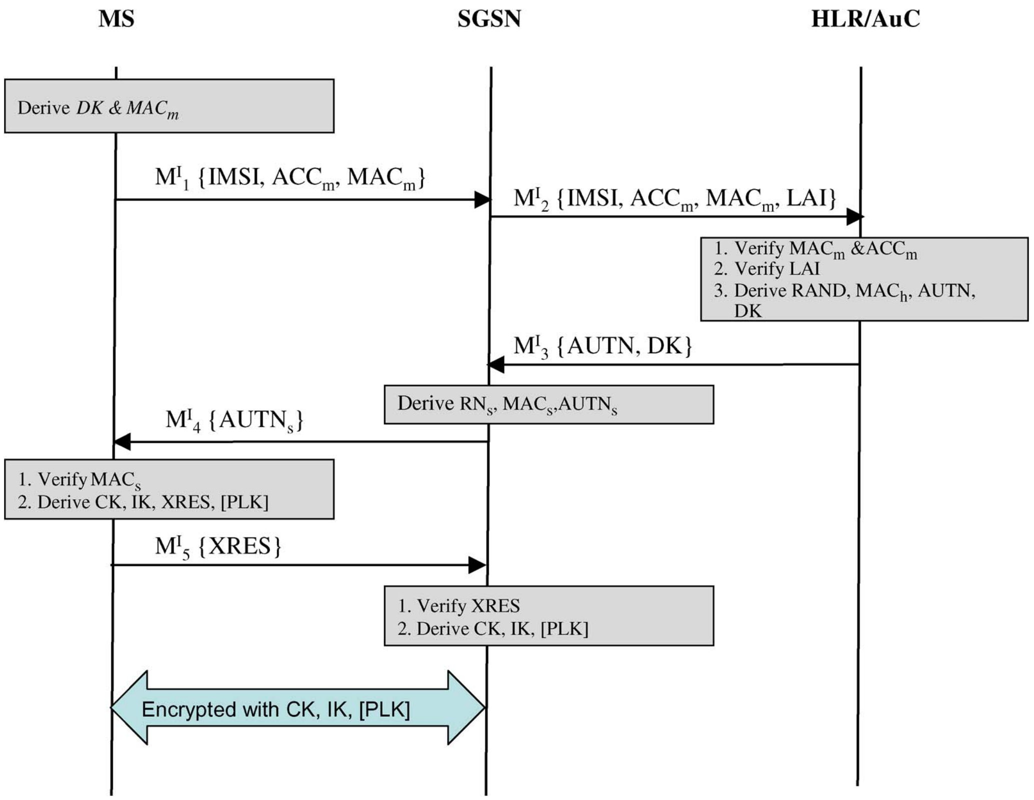
MS
SGSN
HLR/AuC
Derive DK & MAC_m
M^{I}_{1} {IMSI, ACC_m, MAC_m}
M^{I}_{2} {IMSI, ACC_m, MAC_m, LAI}
1. Verify MAC_m &ACC_m
2. Verify LAI
3. Derive RAND, MAC_h, AUTN, DK
M^{I}_{3} {AUTN, DK}
Derive RN_s, MAC_s,$AUTN_s$
M^{I}_{4} {$AUTN_s$}
1. Verify MAC_s
2. Derive CK, IK, XRES, [PLK]
M^{I}_{5} {XRES}
1. Verify XRES
2. Derive CK, IK, [PLK]
Encrypted with CK, IK, [PLK]

Fig. 3. S-AKA-I. The SGSN obtains the authentication vectors from the HLR/AuC.

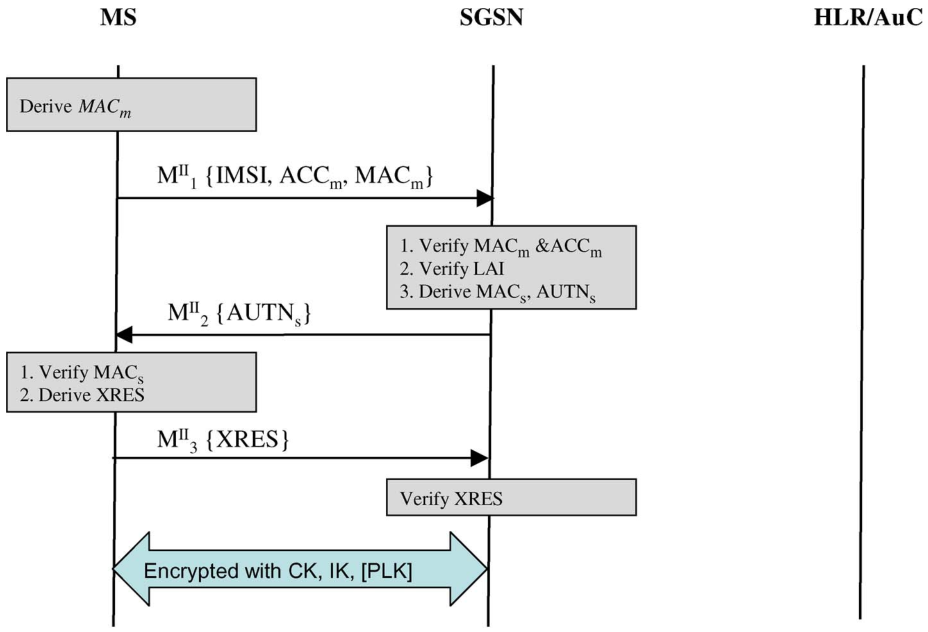

Fig. 4. S-AKA-II. The SGSN mutually authenticates the MS without the involvement of the HLR/AuC.

3) Each MS and its HLR/AuC share a secret key SK and cryptographic functions.

There are two phases defined in S-AKA, namely, S-AKA-I and S-AKA-II. In S-AKA-I, the SGSN intends to obtain AVs from the HLR/AuC, so that the SGSN and MS can authenticate each other without the HLR/AuC in S-AKA-II, as shown in Figs. 3 and 4, respectively.

A. *Phase I: S-AKA-I*

$\mathrm{M}_1^{\mathrm{I}}$ MS → SGSN : {IMSI, Service Request, ACC_m, MAC_m}. In $\mathrm{M}_1^{\mathrm{I}}$, IMSI is the identity of a subscriber. ACC_m presents the number of successful MS authentications and is used to guarantee the freshness of the authentication request. ACC_m, which is initially set to 0, increases on each successful authentication. LAI is the identifier of the location area of the BSS, and it indicates the physical connection between the MS and the BSS. MAC_m is the keyed message authentication code of ACC_m and LAI, protecting the message integrity. It is represented as $\text{MAC}_m = f1_{\text{DK}}(\text{ACC}_m \| \text{LAI})$, where $\text{DK} = f6_{\text{SK}}(\text{ACC}_m)$.

M_2^{I} SGSN → HLR/AuC : $\{\text{IMSI}, \text{ACC}_m, \text{MAC}_m, \text{LAI}\}$. Upon receipt of M_1^{I}, the SGSN records ACC_m. Since the SGSN knows the LAI of the BSS forwarding M_1^{I}, it forwards M_1^{I} to the HLR/AuC together with the BSS's LAI. By checking MAC_m, the HLR/AuC can verify whether the LAI reported by the SGSN is the same as that recognized by the MS. If not, it rejects the request.

M_3^{I} HLR/AuC → SGSN : $\{\text{AUTN}, \text{DK}\}$. The HLR/AuC checks MAC_m for the integrity of ACC_m and LAI. It compares ACC_m and ACC_h counted by the HLR/AuC. The HLR/AuC considers it a replay if $\text{ACC}_m < \text{ACC}_h$. Otherwise, the HLR/AuC randomly generates RAND and derives $\text{MAC}_h = f1_{\text{SK}}(\text{RAND} \| \text{AMF})$, where AMF is an indication of algorithms and keys that generate AVs. Then, the HLR/AuC concatenates the aforementioned tokens to derive $\text{AUTN} = (\text{MAC}_h \| \text{RAND} \| \text{AMF})$. The HLR/AuC also computes DK and sends it to the SGSN together with AUTN. After that, the SGSN successfully obtains the authorization to authenticate the MS for the subsequent connections.

M_4^{I} SGSN → MS : $\{\text{AUTN}_s\}$. Upon receipt of M_3^{I}, the SGSN increments its ACC_s by 1 and randomly generates RN_s, derives MAC_s, and constructs AUTN_s, where $\text{MAC}_s = f1_{\text{DK}}(\text{MAC}_h \| \text{RN}_s \| \text{RAND} \| \text{ACC}_s)$, and $\text{AUTN}_s = \text{MAC}_s \| \text{RN}_s \| \text{RAND} \| \text{AMF} \| \text{ACC}_s$.

M_5^{I} MS → SGSN : $\{\text{XRES}\}$. First, the SGSN checks if $\text{ACC}_s > \text{ACC}_m$ and sets ACC_m to ACC_s when the inequality holds. Second, the MS authenticates the SGSN by deriving and verifying XMAC_h and XMAC_s. Third, the MS computes $\text{IK} = f3_{\text{DK}}(\text{RN}_s)$, $\text{CK} = f4_{\text{DK}}(\text{RN}_s)$, and $\text{XRES} = f2_{\text{DK}}(\text{RN}_s)$ and sends XRES to the SGSN for mutual authentication. If the MS is successfully authenticated, the SGSN uses $f3$ and $f4$, taking DK and RN_s as parameters, to derive CK and IK, respectively. These keys can be used to protect the communication between the MS and the SGSN. As aforementioned, security weaknesses of GSM expose the entire mobile system to man-in-the-middle attacks. If a GSM BSS is involved in a conversation, an extra key $\text{PLK} = f7_{\text{DK}}(\text{RN}_s)$ is negotiated between the MS and the SGSN to protect the confidentiality of the data passing through the GSM BSS.

B. Phase II: S-AKA-II

In this phase, no HLR/AuC is involved. Only three messages are required upon reconnecting to the same SGSN. The SGSN can authenticate the MS according to the AVs obtained in S-AKA-I. The message flow of S-AKA-II is described here.

M_1^{II} MS → SGSN : $\{\text{IMSI}, \text{Service Request}, \text{ACC}_m, \text{MAC}_m\}$. Similar to M_1^{I}, the MS increments its ACC_m by 1 and sends $\text{MAC}_m = f1_{\text{DK}}(\text{ACC}_m \| \text{LAI})$ to the SGSN. ACC_m continues from the ACC_m in the previous M_1^{I}. In addition, SK in M_1^{I} is replaced with DK for there is no preshared keys between the MS and the SGSN.

M_2^{II} SGSN → MS : $\{\text{AUTN}_s\}$. The SGSN checks the LAI of the BSS. If the BSS is not physically connected, the SGSN rejects the request immediately. Otherwise, the SGSN accumulates the ACC_s by 1 and compares it with the ACC_m of M_1^{II} to check if it is a replay. The SGSN then verifies MAC_m on behalf of the HLR/AuC. If MAC_m is legitimate, the SGSN generates RN_s and computes $\text{MAC}_s = f1_{\text{DK}}(\text{MAC}_h \| \text{RN}_s \| \text{RAND} \| \text{ACC}_s)$, where MAC_h, RAND, and DK are sent by the HLR/AuC. The SGSN then constructs and sends AUTN_s to the MS, where $\text{AUTN}_s = \text{MAC}_s \| \text{RN}_s \| \text{RAND} \| \text{AMF} \| \text{ACC}_s$.

M_3^{II} MS → SGSN : $\{\text{XRES}\}$. Upon receipt of AUTN_s, the MS authenticates the SGSN and HLR/AuC by verifying MAC_s and MAC_h, respectively. Then, the MS sends $\text{XRES} = f2_{\text{DK}}(\text{RN}_s)$ to the SGSN. The SGSN authenticates the MS by verifying the freshness and correctness of XRES.

For each successful authentication, the SGSN increments ACC_s and forwards the new ACC_s to the MS (see M_4^{I}). Meanwhile, the MS sets the new ACC_s to its ACC_m (see M_5^{I}) for synchronizing ACC_m and ACC_s. The synchronized ACC_m and ACC_s can be used to detect potential DoS attacks initiated by forging M_1^{II} in S-AKA-II.

The major enhancements of S-AKA include three factors.

1) *Resistance to the redirection attacks:* In UMTS AKA, LAI, which identifies the location area of the BSS, is not protected and can be altered by an adversary with some redirection attack. S-AKA uses message authentication code to protect the integrity of LAI, thereby preventing the network from redirection attacks.
2) *Resistance to the man-in-the-middle attacks:* A man-in-the-middle attack can occur while connecting to a GSM BSS. In S-AKA, a new key PLK is negotiated to encrypt payloads between the MS and the SGSN. PLK prevents the communication from being eavesdropped or modified. Since no key generation function for PLK is defined in UMTS AKA, a new function $f7$ is introduced in S-AKA to generate PLK for both the MS and the SGSN.
3) *Reduced bandwidth consumption:* With a ticket-based design, the proposed protocol hence allows the HLR/AuC to authorize the SGSN for subsequent and mutual authentications of the MS. Once the HLR/AuC authenticates the MS successfully, it sends the visited SGSN a delegation key DK for subsequent authentications. Such a design benefits from the traffic reduction between the HLR/AuC and the SGSN and thus greatly reduces bandwidth consumption.

V. Analysis

Since S-AKA retains the framework of UMTS AKA, basic security features, such as data integrity and confidentiality, are inherited. Retaining UMTS AKA also helps S-AKA resist various attacks, such as replay and guessing attacks. This section explains how S-AKA can resist redirection, man-in-the-middle, and DoS attacks and compares UMTS AKA and S-AKA in terms of the bandwidth consumed during the authentication procedures.

A. Security Analysis

- *Mutual authentication between the MS and the HLR/AuC:* In S-AKA-I, the HLR/AuC authenticates the MS by verifying ACC_m and MAC_m on receipt of $\mathrm{M}_2^{\mathrm{I}}$. To authenticate the HLR/AuC, the MS checks AUTN_s received in $\mathrm{M}_4^{\mathrm{I}}$. With MAC_s, RN_s, RAND, AMF, and ACC_s contained in AUTN_s, the MS can derive the following expected authentication codes of the HLR/AuC and SGSN: XMAC_h and XMAC_s, where $\mathrm{XMAC}_s = f1_{\mathrm{DK}}(\mathrm{XMAC}_h \| \mathrm{RN}_s \| \mathrm{RAND} \| \mathrm{AMF} \| \mathrm{ACC}_s)$.

  If XMAC_s is equal to MAC_s, both the HLR/AuC and the SGSN are authenticated. This guarantees mutual authentication between the MS and the HLR/AuC. For subsequent authentications, even when the HLR/AuC is not involved, the MS can still authenticate the HLR/AuC with $\mathrm{M}_2^{\mathrm{II}}$ in S-AKA-II.
- *Mutual authentication between the MS and the SGSN:* In S-AKA-I, the SGSN authenticates the MS by verifying XRES in $\mathrm{M}_5^{\mathrm{I}}$. Upon receipt of $\mathrm{M}_5^{\mathrm{I}}$, the SGSN checks $\mathrm{XRES} = f2_{\mathrm{DK}}(\mathrm{RN}_s)$. The MS is considered authenticated if the equality holds. The same procedure takes place in authenticating the MS when the SGSN receives $\mathrm{M}_5^{\mathrm{II}}$ in S-AKA-II. Similar to authenticating the HLR/AuC, on receiving AUTN_s, the MS computes XMAC_h and XMAC_s to authenticate the SGSN if $\mathrm{XMAC}_h = \mathrm{XMAC}_s$ holds. This ensures mutual authentication between the MS and the SGSN.
- *Freshness of security keys:* In S-AKA-I, CK and IK are negotiated in $\mathrm{M}_4^{\mathrm{I}}$ and $\mathrm{M}_5^{\mathrm{I}}$, whereas in S-AKA-II, they are negotiated in $\mathrm{M}_2^{\mathrm{II}}$ and $\mathrm{M}_3^{\mathrm{II}}$. Since CK and IK are derived from RN_s, the freshness of these keys can be guaranteed by RN_s. ACC_s in $\mathrm{M}_4^{\mathrm{I}}$ or $\mathrm{M}_2^{\mathrm{II}}$ is accumulated on each successful authentication and can be used to guarantee the freshness of $\mathrm{M}_4^{\mathrm{I}}$ and $\mathrm{M}_2^{\mathrm{II}}$. The freshness of RN_s, RAND, and AMF in $\mathrm{M}_4^{\mathrm{I}}$ and $\mathrm{M}_2^{\mathrm{II}}$ can thus be guaranteed as well. This ensures the freshness of CK and IK.

B. Resistance to Attacks

TABLE III
ROBUSTNESS AGAINST DIFFERENT ATTACKS

Attacks	UMTS	AP-	X-	Cocktail-	S-
	AKA				
Redirection	N	Y	N	Y	Y
Man-in-the-middle	N	N	N	Y	Y
Replay	Y	Y	Y	Y	Y
DoS	N	N	P	N	P

Y: Robust; N: Not robust; P: Partially robust

- *Redirection attack:* An adversary initiates a redirection attack by simulating a BSS to obtain user information and by impersonating an MS to forward user messages to his destination. The redirection attack fails if the adversary fails to obtain user information by impersonating a BSS. Without the user information, the adversary cannot impersonate any MS and connect to a legitimate BSS. To impersonate a BSS, the adversary either transmits signals with stronger power or jams the spectrum and tries to entrap the MS to establish the connection with the faked BSS. In S-AKA, the MS embeds the LAI of the BSS in MAC_m and sends MAC_m to the SGSN in $\mathrm{M}_1^{\mathrm{I}}$. The authentication request is rejected if the HLR/AuC fails to match the LAI reported by the SGSN in $\mathrm{M}_2^{\mathrm{I}}$ and the LAI embedded in MAC_m. Such a design not only solves the mischarged billing problems but prevents a user from being tricked into a network with none or weak encryption keys as well.
- *Man-in-the-middle attack:* To defeat the man-in-the-middle attack, an encrypt key PLK is introduced to protect payloads. The key is negotiated by the MS and SGSN after exchanging $\mathrm{M}_4^{\mathrm{I}}, \mathrm{M}_5^{\mathrm{I}}$ and $\mathrm{M}_2^{\mathrm{II}}, \mathrm{M}_3^{\mathrm{II}}$ in S-AKA-I and S-AKA-II, respectively. PLK is used to encrypt and decrypt data only when connecting to a GSM BSS. With PLK, the MS encrypts the payload prior to transmission, even if none encryption command is specified by the GSM BSS. Hence, data confidentiality of the communication channel between the MS and the SGSN can be guaranteed.

  Considering the performance issue, bitwise operations can be used to implement the payload encryption. In general, bitwise encryptions do not consume significant computing power, and data confidentiality can be guaranteed without sacrificing performance.
- *DoS attack:* During the initial authentication, a malicious MS may launch a DoS attack either to its HLR/AuC (using $\mathrm{M}_1^{\mathrm{I}}$) or to the visited SGSN (using $\mathrm{M}_1^{\mathrm{II}}$).

  - If the MS forges $\mathrm{M}_1^{\mathrm{I}}$, the forged message can be detected by the HLR/AuC on receipt of $\mathrm{M}_2^{\mathrm{I}}$.
  - If the MS forges $\mathrm{M}_1^{\mathrm{II}}$, the forged message can be immediately detected by the SGSN with DK authorized by the HLR/AuC.

  We claim that S-AKA can partially resist DoS attacks since the forged $\mathrm{M}_1^{\mathrm{II}}$ can be immediately detected by the SGSN but the forged $\mathrm{M}_1^{\mathrm{I}}$ can only be detected by the HLR/AuC.

  Table III lists a summary of robustness against known attacks to the AKA protocols proposed for UMTS networks. Most of the AKA protocols (marked N) fail to detect forged messages at the SGSN side during the initial authentication, but some of them (marked P) can detect the forged messages during subsequent authentications.

C. Bandwidth Consumption

In analyzing the bandwidth consumption, we assume that m AVs are transmitted every time the HLR/AuC successfully authenticates the MS. We also assume that the MS averagely issues p authentication requests to the same SGSN.

1) Bandwidth Analysis of UMTS AKA: The sizes of UM_1 to UM_5 are calculated as follows.

- The length of the first message, which is denoted by $|\mathrm{UM}_1|$, is the sum of the length of its parameters: IMSI, Service Request, and LAI. Thus

$$
\begin{aligned}
|\mathrm{UM}_1| &= |\mathrm{IMSI}| + |\text{Service Request}| + |\mathrm{LAI}| \\
&= 176 \text{ bits.}
\end{aligned}
$$

- Since UM_2 is a forwarding message of UM_1, its length is the same as UM_1.
- UM_3 contains a sequence of AV, which is composed of RAND, XRES, CK, IK, and AUTN. Its length can be represented as

$$
\begin{aligned}
|\mathrm{AV}| &= |\mathrm{RAND}| + |\mathrm{XRES}| + |\mathrm{CK}| + |\mathrm{IK}| + |\mathrm{AUTN}| \\
&= 608 \text{ bits.}
\end{aligned}
$$

Since m AVs are assumed to be transmitted by the HLR/AuC after the initial authentication, the length of UM_3 is $m * |\mathrm{AV}| = 608m$ bits.

- UM_4 consists of RAND and $\mathrm{AUTN} = (\mathrm{SQN} \oplus \mathrm{AK} \| \mathrm{AMF} \| \mathrm{MAC})$. Its length can be obtained by summing up $|\mathrm{RAND}| + |\mathrm{AUTN}|$ (288 bits).
- UM_5 only contains a 64-bit long RES.

In UMTS AKA, the bandwidth consumption varies depending on whether the HLR/AuC is involved or not. The HLR/AuC is required to authenticate the MS if there is no available AV in the SGSN. No HLR/AuC is required if there is any unused AV in the SGSN. The bandwidth consumption for these two cases is discussed as follows.

- *No available AV in the SGSN:* Since five messages are exchanged, the bandwidth consumption is obtained by summing up the lengths of the five messages, i.e.,

$$
\mathrm{bw}_{\mathrm{init}} = \sum_{i=1}^{5} |\mathrm{UM}_i| = 704 + 608m \text{ bits.}
$$

- *Available AVs in the SGSN:* In this case, only UM_1, UM_4, and UM_5 are exchanged between the MS and the SGSN. The bandwidth consumption is thus

$$
\mathrm{bw}_{\mathrm{sub}} = |\mathrm{UM}_1| + |\mathrm{UM}_4| + |\mathrm{UM}_5| = 528 \text{ bits.}
$$

The overall bandwidth consumption for p times of authentications in UMTS AKA is summarized as

$$
\left\lceil \frac{p}{m} \right\rceil * \mathrm{bw}_{\mathrm{init}} + \left(p - \left\lceil \frac{p}{m} \right\rceil \right) * \mathrm{bw}_{\mathrm{sub}}.
$$

Furthermore, the total number of message exchange is

$$
\left\lceil \frac{p}{m} \right\rceil * 5 + \left(p - \left\lceil \frac{p}{m} \right\rceil \right) * 3, \text{ where } p \geqslant 1 \text{ and } m \geqslant 1.
$$

2) Bandwidth Analysis of S-AKA: The lengths of S-AKA messages are calculated as follows.

- $\mathrm{M}_1^{\mathrm{I}}$ is composed of IMSI, Service Request, ACC_m, and MAC_m. $|\mathrm{M}_1^{\mathrm{I}}|$ is 264 bits.
- $\mathrm{M}_2^{\mathrm{I}}$ appends LAI to $\mathrm{M}_1^{\mathrm{I}}$, and its length is $|M_1^I| + |\mathrm{LAI}| =$ 264 bits.
- $\mathrm{M}_3^{\mathrm{I}}$ contains AUTN and DK, and its length is $|\mathrm{AUTN}| + |\mathrm{DK}| = 368$ bits.
- In $\mathrm{M}_4^{\mathrm{I}}$, the SGSN sends AUTN_s to the MS. Thus, its length should be $|\mathrm{AUTN}_s| = 392$ bits.

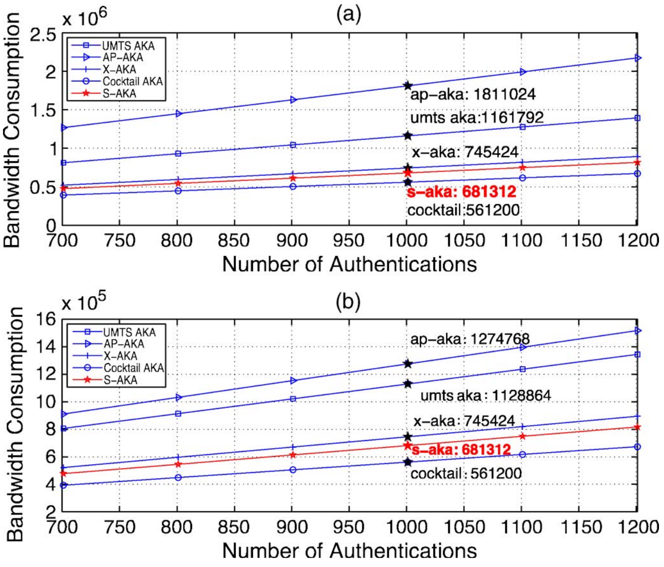

Fig. 5. Bandwidth consumption of UMTS AKA, AP-AKA, X-AKA, Cocktail-AKA, and S-AKA. (a) $m = 2$. (b) $m = 100$. p ranges from 1 to 1200.

- $\mathrm{M}_5^{\mathrm{I}}$ is an expected response XRES.
- In S-AKA-II, $\mathrm{M}_1^{\mathrm{II}}$, $\mathrm{M}_2^{\mathrm{II}}$, and $\mathrm{M}_3^{\mathrm{II}}$ are identical to $\mathrm{M}_1^{\mathrm{I}}$, $\mathrm{M}_4^{\mathrm{I}}$, and $\mathrm{M}_5^{\mathrm{I}}$, respectively. Hence, we obtain

$$
\left| M_1^{II} \right| = \left| M_1^{I} \right|, \quad \left| M_2^{II} \right| = \left| M_4^{I} \right|, \quad \left| M_3^{II} \right| = \left| M_5^{I} \right|.
$$

Similar to UMTS AKA, the bandwidth consumption varies on S-AKA-I and S-AKA-II.

- For an initial authentication (S-AKA-I)

$$
\mathrm{bw}_{\mathrm{init}} = \sum_{i=1}^{5} \left| M_i^{I} \right| = 1312 \text{ bits.}
$$

- For a subsequent authentication (S-AKA-II)

$$
\mathrm{bw}_{\mathrm{sub}} = \sum_{i=1}^{3} \left| M_i^{II} \right| = 680 \text{ bits.}
$$

Therefore, we derive the overall bandwidth consumption for p times of authentications, i.e.,

$$
\mathrm{bw}_{\mathrm{init}} + (p-1) * \mathrm{bw}‘_{\mathrm{sub}}, \text{ for } p \geqslant 1.
$$

In addition, the number of messages exchanged for p times of authentications is

$$
5 + (p-1) * 3, \text{ for } p \geqslant 1.
$$

3) Comparisons: The bandwidth consumption varies by the number of AVs transmitted from the HLR/AuC to the SGSN and the total number of authentication requests. In Fig. 5, we compare UMTS AKA, AP-AKA, X-AKA, Cocktail-AKA, and S-AKA in terms of the number of AVs ($m = 2$ and 100) and the number of authentication requests p. The x-axis stands for the number of authentications within the same SGSN territory, and the y-axis represents the bandwidth consumption (in bits).

Table IV(a) and (b) shows the average ratios of bandwidth consumption and the number of messages exchanged for user

TABLE IV
RATIOS OF BANDWIDTH CONSUMPTION AND MESSAGE EXCHANGE FOR AUTHENTICATION

(a)

m	2	5	10	20	50	100	Avg.
$\frac{\textit{AP–AKA}}{\textit{UMTS AKA}}$	1.56	1.30	1.21	1.17	1.14	1.13	1.25
$\frac{\textit{X–AKA}}{\textit{UMTS AKA}}$	0.64	0.67	0.68	0.69	0.69	0.69	0.68
$\frac{\textit{Cocktail–AKA}}{\textit{UMTS AKA}}$	0.48	0.51	0.51	0.52	0.52	0.52	0.51
$\frac{\textit{S–AKA}}{\textit{UMTS AKA}}$	0.59	0.61	0.62	0.63	0.63	0.63	0.62

(b)

m (p=1000)	2	5	10	20	50	100	Avg.
$\frac{\textit{AP–AKA}}{\textit{UMTS AKA}}$	1.25	1.12	1.06	1.03	1.01	1.01	1.08
$\frac{\textit{X–AKA}}{\textit{UMTS AKA}}$	0.75	0.88	0.94	0.97	0.99	0.99	0.92
$\frac{\textit{Cocktail–AKA}}{\textit{UMTS AKA}}$	0.75	0.88	0.94	0.97	0.99	0.99	0.92
$\frac{\textit{S–AKA}}{\textit{UMTS AKA}}$	0.75	0.88	0.94	0.97	0.99	0.99	0.92

authentications. In Table IV(a), the average of bandwidth ratios (for $m = 2$, 5, 10, 20, 50, and 100) for S-AKA/UMTS AKA is 0.62, which means S-AKA has reduced 38% of the bandwidth. Similarly, the average of message ratios (S-AKA/UMTS AKA) in Table IV(b) shows that S-AKA has reduced 8% of the messages exchanged for authentication. Despite that S-AKA is not the protocol that saves the most bandwidth, it can resist more attacks, as described in Table III.

VI. FORMAL ANALYSIS

Traditional formal logics were developed to find protocol flaws, but they do not appear to provide security guarantees used in analyzing higher level protocols using session keys. In 1999, Shoup [19] proposed a new formal security model specifying security guarantees that a session key exchange protocol should provide. Taking Shoup's model as a basis, Zhang [20] proposed a security model consisting of an ideal and a real system to analyze simulatability of adversaries in the two systems and examine the security for key exchange protocols.

In this paper, we utilize Shoup's and Zhang's [20] formal models to analyze AKA protocols in the mobile settings. We identify the following two types of communication channels of mobile networks: 1) channels within and between networks and 2) channels between users and networks. In practice, channels of the former type are protected through dedicated communication circuits or high-layer security schemes. Channels of the latter type are usually implemented using wireless media and, thus, are vulnerable to attacks.

We assume that an adversary is capable of fully controlling channels between users and networks, including *eavesdropping*, *modifying*, and *replaying* intercepted messages. In the following, we specify the actions of adversaries for both ideal and real systems defined in Shoup's security model. The ideal system describes the authentication between user entities and network entities. It can be treated in the same way as in the two-party setting defined in Shoup's formal model of security. The real system models the operations executed by a real-world adversary who controls the communication channels between a user and a network. It thus follows the definition of the three-party setting in Shoup's security model. The security of an AKA protocol can be proved based on the simulatability in the two different systems.

A. Preliminaries

We summarize the definitions of advantages presented in Zhang's model [3], [20] as follows.

- The distinguishing advantage of a probabilistic polynomial-time algorithm D that outputs 0 or 1 is defined as $\mathrm{Adv}^{\mathrm{dist}}_{x_k,y_k}(D) = |P(D(x_k) = 1) - P(D(y_k) = 1)|$, where $x = \{x_k\}_{k \geqslant 0}$ and $y = \{y_k\}_{k \geqslant 0}$ are sequences of random variables; x_k and y_k are in a finite set.
- The prf advantage of a probabilistic polynomial-time oracle machine A is defined as $\mathrm{Adv}^{\mathrm{prf}}_{G}(A) = |P(g \overset{R}{\leftarrow} G : A^g = 1) - P(g \overset{R}{\leftarrow} U(d,s) : A^g = 1)|$, where $g \overset{R}{\leftarrow} G$ denotes the operation of randomly selecting a function g from the family $G : \{0,1\}^k \times \{0,1\}^d \to \{0,1\}^s$, and $U(d,s)$ denotes the family of all functions from $\{0,1\}^d$ to $\{0,1\}^s$. G can be also associated with an insecurity function $\mathrm{Adv}^{\mathrm{prf}}_{G}(t,q) = A \overset{\mathrm{MAX}}{\in} A(t,q)\ \mathrm{Adv}^{\mathrm{prf}}_{G}(A)$, where $A(t,q)$ denotes the set of adversaries that make at most q oracle queries and have running time at most t.
- The mac advantage of an adversary A, i.e., $\mathrm{Adv}^{\mathrm{mac}}_{F}(A)$, is defined as the probability that $A^{F(K)}$ outputs a pair (σ, M), where we have the following.
  - $F : \{0,1\}^k \times \mathrm{Dom}(F) \to \{0,1\}^l$ is a family of functions generating MAC, where $\mathrm{Dom}(F) = \{0,1\}^{\leqslant L}$.
  - $K \in \{0,1\}^k$ is a randomly chosen key.
  - M was not a query of A to its Oracle.
  - $\sigma = F(K,M)$ is referred to as the MAC of M.

  F can be associated with an insecurity function $\mathrm{Adv}^{\mathrm{mac}}_{F}(t,q) = A \overset{\mathrm{MAX}}{\in} A(t,q)\ \mathrm{Adv}^{\mathrm{mac}}_{F}(A)$, where $A(t,q)$ denotes the set of adversaries that make at most q oracle queries and have running time at most t. If $\mathrm{Adv}^{\mathrm{mac}}_{F}(A)$ is negligible in k for every polynomially bounded adversary A, we say that F is a secure MAC.

B. Security Proofs

By [20, Definitions 1 and 2], we assume that each entity in S-AKA has a random number generator producing random numbers, such as RAND and RN_s, for the network entity and its instances. In addition, we assume these random numbers are randomly selected in the game of A, and $|\mathrm{RAND}|$ and $|\mathrm{RN}_s|$ are polynomials in k. Let C_A denote the event that the transcript of A (T_A) is collision free, and let $\overline{C_A}$ be the complement of event C_A. Then, we derive the probability of $\overline{C_A}$, i.e.,

$$P(\overline{C_A}) \leqslant \frac{n_i^2\left(2^{-|\mathrm{RAND}|} + 2^{-|\mathrm{RN}_s|}\right)}{2} \tag{1}$$

where n_i denotes the number of instances initialized by A. Since $|\text{RAND}|$ and $|\text{RN}_s|$ are assumed polynomials in k, we consider that $P(\overline{C_A})$ is negligible.

Lemma 1: Let A be a real-world adversary, and let T_A be the transcript of A, which is assumed collision free. Assuming $f1$ and $f2$ are independent function families and collision resistant in T_A, let M_A denote the event that T_A is authentic. Then, we obtain the probability of its complement event, i.e.,

$$P(\overline{M_A}) \leqslant n_i \left(2 * \text{Adv}_F^{\text{mac}}(t, q)\right). \tag{2}$$

Proof: If T_A were not authentic, there must have existed at least one instance which has been 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 (2). The proof considers three cases.

1) Let $I_{i'j'}$ be the network instance that has received and accepted $(\text{IMSI}, \text{ACC}_m, \text{MAC}_m, \text{LAI})$. Since $\text{IMSI}_{i'j'}$ is used in the computation of MAC_m, the stimulus on $I_{i'j'}$ could not be output by any user instance that is not compatible with $I_{i'j'}$. We can then construct an adversary A_F for message authentication code F. A_F has oracle access to $f1_K$ and $f2_K$, where K was randomly chosen. Assume that $\text{IMSI}_{i'j'}$ is assigned to a user U, which may or may not be initialized by A. A_F begins its experiment by selecting the authentication keys for all users but U. 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 oracles $f1_K$ and $f2_K$. If an entity or entity instance needs to evaluate $f3$, $f4$, $f6$, and $f7$ 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 $(\text{MAC}_m, \text{ACC}_m \| \text{LAI})$; otherwise, A_F stops at the end of the game of A and outputs an empty string.

   Let $\text{Succ}(A_F, F)$ denote the event that A_F outputs a MAC_m and a message, and the message was not queried to oracle $f1_K$. Let $\text{AS}_{i'j'}$ denote the event that $I_{i'j'}$ has been accepted, but the stimulus on $I_{i'j'}$ was not output by a user instance. If $\text{AS}_{i'j'} = 1$, then A_F has successfully forged MAC_m for message $\text{ACC}_m \| \text{LAI}$, and this message was not queried to oracle $f1_k$. This implies $P(\text{AS}_{i'j'} = 1) \leqslant P(\text{Succ}(A_F, F)) = 1$. Thus, we can obtain

$$P(\text{AS}_{i'j'} = 1) \leqslant \text{Adv}_F^{\text{mac}}(t, q) \tag{3}$$

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

2) Let I_{ij} be a user instance that has received and accepted AUTN_s. Let AS_{ij} denote the event that the stimulus on I_{ij} was not output by any network instance. Let IS_{ij} denote the event that the stimulus on I_{ij} was output by any network instance $I_{p'q'}$ but not compatible with I_{ij}. If IS_{ij} is true, then instance $I_{p'q'}$ received message $(\text{IMSI}, \text{ACC}, \text{MAC}_m, \text{LA}I)$ before sending out AUTN_s. Since T_A is collision free, RN_s and RAND cannot be generated by any other user instances except I_{ij}. This implies that A has successfully concocted MAC_m. By (3), we have

$$P(\text{IS}_{ij} = 1) \leqslant \text{Adv}_F^{\text{mac}}(t, q) \tag{4}$$

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

   Now, suppose AS_{ij} is true, then adversary A has successfully concocted MAC_h and MAC_s. By running A, we can construct an adversary A'_F for $f1$. A'_F works in the same way as $f1$ except that, when I_{ij} accepts, A'_F stops and outputs the following two pairs: $(\text{MAC}_h, \text{RAND}\|\text{AMF})$ and $(\text{MAC}_s, \text{MAC}_h\|\text{RN}_s\|\text{ACC}_s\|\text{RAND})$. Using the notation $\text{Succ}(A'_F, F)$ described earlier, we have

$$P(\text{AS}_{ij} = 1) \leqslant P\left(\text{Succ}(A_F, F) = 1\right). \tag{5}$$

   Therefore, by (4) and (5), the probability of the stimulus on a user instance I_{ij} that was not output by a compatible network instance is upper bounded by $P(\text{AS}_{ij} = 1) + P(\text{IS}_{ij=1}) \leqslant 2 * \text{Adv}_F^{\text{mac}}(t, q)$.

3) Let $I_{i''j''}$ be a network instance that has received and accepted XRES, where RN_s in AUTN_s was sent by $I_{i''j''}$. If the stimulus on $I_{i''j''}$ was not output by any user instance, then adversary A has successfully concocted XRES. Similar to (3), it is proved that the probability of such an event is upper bounded by $\text{Adv}_F^{\text{mac}}(t, q)$.

   Next, if the stimulus on $I_{i''j''}$ was output by a user instance I_{pq} that is not compatible with $I_{i''j''}$, then I_{pq} received AUTN_s before it outputs the stimulus. Since T_A is collision free, AUTN_s cannot be output by any network instance other than $I_{i''j''}$. This means that the adversary concocted MAC_s for $(\text{MAC}_h\|\text{RN}_s\|\text{ACC}_s\|\text{RAND})$. By (5), the probability of such an event is upper bounded by $2 * \text{Adv}_F^{\text{mac}}(t, q)$.

We then conclude that the probability that T_A is not an authentic transcript is at most $n_i(2 * \text{Adv}_F^{\text{mac}}(t, q))$, where n_i is the number of instances. ■

Lemma 2: Let A be a real-world adversary and T_A be the transcript of A. Assume that T_A is authentic and collision free and G is a pseudorandom function family, independent of $f1$, where $f1$ is collision resistant in T_A. Then, there exists an ideal-world adversary A^* such that, for every distinguisher D with running time T, $\text{Adv}_{T_A, T_A^*}^{\text{dist}}(D) \leqslant n_e \text{Adv}_G^{\text{prf}}(t, q)$, where n_e and n_i are the numbers of user entities and instances initialized by A, respectively, $t = O(T)$, and $q = O(n_i)$.

Proof: We construct a simulator that takes a real-world adversary A as the input and creates an ideal-world adversary A^*. The simulator basically has A^* acting as 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.

- For each record (**start session**, i, j) that A's action cause is in the real-world transcript, A^* computes a connection assignment, and the ringmaster in the ideal system substitutes session key K_{ij}^s with an idealized random session key K_{ij}.

- For each record (**abort session**, i, j) that A's action cause is 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. Thus, we have an ideal-world adversary whose transcript is almost identical to the transcript of real-world adversary A. The differences exist in the application records. In the following three cases, 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 $I_{i_1 j_1}$ has received and accepted message AUTN$_s$. Since T_A is authentic, this message must be output by a network instance $I_{i'_1 j'_1}$ compatible with $I_{i_1 j_1}$. In this case, we let adversary A^* make the connection assignment (**create**, i'_1, j'_1). We have to argue that this connection assignment was never made before. The truth holds because AUTN$_s$ could not be a stimulus on other user instances. Otherwise, MAC$_s$ would not be acceptable by $I_{i_1 j_1}$. Therefore, it is legal for A^* to make the connection assignment. Consequently, it is also legal to substitute session key $K^s_{i_1 j_1}$ with a random number $K_{i_1 j_1}$.
- *Case 2:* Assume that a network instance $I_{i'_2 j'_2}$ has received and accepted message (IMSI, ACC$_m$, MAC$_m$, LAI) from a user instance $I_{i_2 j_2}$. We let A^* make the connection assignment (**create**, i_2, j_2) and let the ringmaster substitute session key $K^s_{i'_2 j'_2}$ with a random number $K_{i'_2 j'_2}$. Since $f1$ is collision resistant in T_A, MAC$_m$ could not be a stimulus on any instances other than $I_{i'_2 j'_2}$. Therefore, the connection assignment (**create**, i_2, j_2) cannot be made before.
- *Case 3:* Assume that a network instance $I_{i'_3 j'_3}$ has received and accepted message XRES. Under the assumption that T_A is collision free and $f2$ is collision resistant in T_A, it can be concluded that $I_{i_3 j_3}$ has been accepted and the stimulus on $I_{i_3 j_3}$ output by $I_{i'_3 j'_3}$. By Case 1, $I_{i_3 j_3}$ has been isolated by $I_{i'_3 j'_3}$. It is legal for A^* to make the connection assignment (**connect**, i_3, j_3). Accordingly, the ringmaster replaces session key $K_{i'_3 j'_3}$ with $K_{i_3 j_3}$.

The aforementioned 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. Therefore, we only need to consider the application records in both transcripts.

First, we 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 $\mathrm{Adv}^{\mathrm{dist}}_{T_A, T^*_A}(D) = \mathrm{Adv}^{\mathrm{prf}}_{G}(D')$. Thus, $\mathrm{Adv}^{\mathrm{dist}}_{T_A, T^*_A}(D) \leqslant \mathrm{Adv}^{\mathrm{prf}}_{G}(t, q)$, where $t = O(T), q = O(2n_i)$, and n_i is the number of instances initialized by A. Now, assume the number of user entities initialized by A in n_e. Let K_1, K_2, …, and K_{n_e} denote the keys of the user entities. Then, D and D' have access to the input-and-output pairs of $G_{K_1}, G_{K_2}, \ldots, G_{K_e}$. As a result, it can be concluded that

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

which proves the lemma. ■

Theorem VI.1: If G is a pseudorandom function family, $f1$ is a secure message authentication code, and G and $f1$ are independent, then S-AKA is an S-AKA protocol.

Proof: Let A be a real world adversary and 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, we 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

$$\begin{aligned} &\left|P\left(D(T_A) = 1 | M_A \cap C_A\right) - P\left(D(T_{A^*}) = 1 | M_A \cap C_A\right)\right| \\ &\qquad\qquad\qquad\qquad \leqslant n_e \mathrm{Adv}^{\mathrm{prf}}_{G}(t, q). \end{aligned}$$

Thus, it follows that

$$\begin{aligned} &\mathrm{Adv}^{\mathrm{dist}}_{T_A, T^*_A}(D) \\ &\quad = \left|P\left(D(T_A) = 1\right) - P\left(D(T_{A^*}) = 1\right)\right| \\ &\quad = \left|\left(P\left(D(T_A) = 1 | M_A \cap C_A\right)\right.\right. \\ &\qquad\quad \left.\left. - P\left(D(T_{A^*}) = 1 | M_A \cap C_A\right)\right) P\left(M_A \cap C_A\right)\right| \\ &\qquad + \left|\left(P\left(D(T_A) = 1 | \overline{M_A} \cup \overline{C_A}\right)\right.\right. \\ &\qquad\qquad \left.\left. - P\left(D(T_{A^*}) = 1 | \overline{M_A} \cup \overline{C_A}\right)\right) P\left(\overline{M_A} \cup \overline{C_A}\right)\right| \\ &\quad \leqslant \left|P\left(D(T_A) = 1 | M_A \cap C_A\right)\right. \\ &\qquad\quad \left. - P\left(D(T_{A^*}) = 1 | M_A \cap C_A\right)\right| + P(\overline{M}_A) + P\left(\overline{C}_A\right) \\ &\quad \leqslant n_e \mathrm{Adv}^{\mathrm{prf}}_{G}(t, q) + P(\overline{M}_A) + P(\overline{C}_A) \end{aligned}$$

and we obtain

$$\begin{aligned} P(\overline{M_A}) &= P(\overline{M_A} | C_A) P(C_A) + P(\overline{M_A} | \overline{C_A}) P(\overline{C_A}) \\ &\leqslant P(\overline{M_A} | C_A) + P(\overline{C_A}). \end{aligned}$$

Now, we conclude that

$$\mathrm{Adv}^{\mathrm{dist}}_{T_A, T^*_A}(D) \leqslant n_e \mathrm{Adv}^{\mathrm{prf}}_{G}(t, q) + P(\overline{M_A} | C_A) + 2P(\overline{C_A}).$$

By (1), $P(\overline{C_A})$ is negligible in k. In addition, by Lemma 1, $P(\overline{M_A} | C_A)$ is also negligible. Hence, $\mathrm{Adv}^{\mathrm{dist}}_{T_A, T^*_A}(D)$ can be considered negligible. This has proved that S-AKA is an S-AKA protocol. ■

VII. Conclusion

To resolve the vulnerabilities found in GSM systems, UMTS AKA was designed to defeat many known security issues and has been adopted in 3G/4G networks for securely authenticating mobile subscribers. Despite the security enhancement, UMTS AKA is still vulnerable to some attacks, such as redirection and man-in-the-middle attacks. In this paper, we have analyzed the security weakness of UMTS AKA and proposed a new authentication key agreement protocol, namely, S-AKA, for UMTS networks.

The proposed protocol is more efficient and can defeat both redirection and man-in-the-middle attacks. We have also analyzed the message exchange and bandwidth consumption of S-AKA and compared it with UMTS AKA. The result shows that, in terms of bandwidth consumption, our protocol can save up to 38% of the bandwidth required during authentication. In addition, we have formally proved the security strength and robustness of our protocol using Shoup's and Zhang's formal models.

REFERENCES

[1] A. Peinado, "Privacy and authentication protocol providing anonymous channels in GSM," *Comput. Commun.*, vol. 27, no. 17, pp. 1709–1715, Nov. 2004.

[2] B. S. Babu and P. Venkataram, "A dynamic authentication scheme for mobile transactions," *Int. J. Netw. Secur.*, vol. 8, no. 1, pp. 59–74, Jan. 2009.

[3] M. Zhang, "Provably-secure enhancement on 3GPP authentication and key agreement protocol," Verizon Commun., Cryptology ePrint Archive Rep. 2003/092, 2003.

[4] U. Meyer and S. Wetzel, "A man-in-the-middle attack on UMTS," in *Proc. 3rd ACM WiSe*, New York, 2004, pp. 90–97.

[5] Technical Specification Group Services and System Aspects; 3G Security; Security Architecture, Third Generation Partnership Project, Tech. Rep. Tech. Spec. 3G TS 33.102 V3.7.0, 2000.

[6] M. Zhang and Y. Fang, "Security analysis and enhancements of 3GPP authentication and key agreement protocol," *IEEE Trans. Wireless Commun.*, vol. 4, no. 2, pp. 734–742, Mar. 2005.

[7] K.-M. Cheng, T.-Y. Chang, and J.-W. Lo, "Cryptanalysis of security enhancement for a modified authentication key agreement protocol," *Int. J. Netw. Secur.*, vol. 11, no. 1, pp. 55–57, Jul. 2010.

[8] C. C. Chang, K. F. Hwang, and I. C. Lin, "Security enhancement for a modified authenticated key agreement protocol," *Int. J. Comput. Numer. Anal. Appl.*, vol. 3, no. 1, pp. 1–7, 2003.

[9] D. Seo and P. Sweeney, "Simple authenticated key agreement algorithm," *Electron. Lett.*, vol. 35, no. 13, pp. 1073–1074, Jun.1999.

[10] S. I. Gy. Gdor, "Novel authentication algorithm public key based cryptography in mobile phone systems," *Int. J. Comput. Sci. Netw. Secur.*, vol. 6, no. 2B, pp. 126–134, Feb. 2006.

[11] J. A. Murtaza Naveed Akhtar and A. Ali Minhas, "A novel security algorithm for universal mobile telecommunication system," *Int. J. Multimedia Ubiquitous Eng.*, vol. 5, no. 1, pp. 1–18, Jan. 2010.

[12] Y.-B. Lin, M.-F. Chang, M.-T. Hsu, and L.-Y. Wu, "One-pass GPRS and IMS authentication procedure for UMTS," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 6, pp. 1233–1239, Jun. 2005.

[13] A. I. Gardezi, "Security in wireless cellular networks," Washington University in St. Louis, St. Louis, MO, 2006.

[14] C.-M. Huang and J.-W. Li, "Authentication and key agreement protocol for UMTS with low bandwidth consumption," in *Proc. 19th Int. Conf. AINA*, 2005, pp. 392–397.

[15] J. Al-Saraireh and S. Yousef, "A new authentication protocol for UMTS mobile networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2006, no. 2, p. 19, Apr. 2006.

[16] E. Chun-I, P.-H. Ho, and H.-Y. Chen, "Nested one-time secret mechanisms for fast mutual authentication in mobile communications," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2007, pp. 2714–2719.

[17] H.-H. Ou, M.-S. Hwang, and J.-K. Jan, "A cocktail protocol with the authentication and key agreement on the UMTS," *J. Syst. Softw.*, vol. 83, no. 2, pp. 316–325, Feb. 2010.

[18] S. Wu, Y. Zhu, and Q. Pu, "Security analysis of a cocktail protocol with the authentication and key agreement on the UMTS," *Commun. Lett.*, vol. 14, no. 4, pp. 366–368, Apr. 2010.

[19] V. Shoup, "On formal models for secure key exchange," IBM Zurich Research Lab, Rüschlikon, Switzerland, Tech. Rep. RZ 3120 (#93166), 1999.

[20] M. Zhang, "Adaptive protocol for entity authentication and key agreement in mobile networks," in *Proc. ICISC*, 2003, pp. 166–183.

Yu-Lun Huang (M'04) received the B.S. and Ph.D. degrees in computer science and information engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1995 and 2001, respectively.

She is currently an Assistant Professor with the Department of Electrical Engineering, National Chiao Tung University. Her research interests include wireless security, secure testbed design, embedded software, embedded operating systems, risk assessment, secure payment systems, voice over Internet Protocol, and quality of service.

Dr. Huang is a member of the Phi Tau Phi Society.

Chih-Ya Shen received the B.S. and M.S. degrees from National Chiao Tung University, Hsinchu, Taiwan, in 2005 and 2007, respectively. He is currently working toward the Ph.D. degree with the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan.

His research interests include mobile computing and network security.

Shiuhpyng Winston Shieh (SM'98) received the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park.

He is currently a Professor and past Chair of the Department of Computer Science, National Chiao Tung University (NCTU), Hsinchu, Taiwan, and the Director of Taiwan Information Security Center, NCTU. He was a Visiting Professor with the University of California, Berkeley, during 2003–2004 and 2005–2006. He has served as Advisor to the National Security Council of Taiwan Presidential Office, Chair of Malware Forum of National Information and Communication Security Techonology, Director of Government Service Network–Computer Emergency Response Team/Coordination Center, Advisor to the National Information and Communication Security Task Force, and Advisor to the National Security Bureau. He was the former President of the Chinese Cryptology and Information Security Association: one of the leading security organizations in Asia. He is an experimentalist. He (along with V. Gligor of Carnegie Mellon University, Pittsburgh, PA) received the first U.S. patent in the intrusion detection field. He has published more than 150 technical papers, patents, and books. His research interests include reliability and security hybrid mechanisms, network and system security, and software program behavior analysis.

Dr. Shieh has been actively involved with the IEEE Reliability Society, where he serves as the Editor-in-Chief of the IEEE RELIABILITY SOCIETY NEWSLETTER; an Administrative Committee Member and Associate Editor of the IEEE TRANSACTIONS ON RELIABILITY; the Program Chair for the 2012 IEEE Software Security and Reliability; and the Chair of the IEEE Reliability Society Taipei/Tainan Chapter. During his term as the Chapter Chair, the chapter received the Best Chapter Award from both the Reliability Society and the IEEE Taipei Section (among the 41 chapters in the Taipei Section), respectively. In addition, he is an Association for Computing Machinery (ACM) Special Interest Group on Security, Audit and Control Awards Committee Member and the Associate Editor of the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING; a past Associate Editor of the *ACM Transactions on Information and System Security*, *Journal of Computer Security*, *Journal of Information Science and Engineering*, and *Journal of Computers*; and a Guest Editor of IEEE INTERNET COMPUTING. He was one of the 41 recipients worldwide of the ACM Distinguished Scientist Award in 2010. He was also a recipient of the ACM Service Award for his contribution to ACM and the Distinguished Information Award (presented by Taiwan's Vice President) for his contribution to computer security research, which is the highest honor awarded to computer scientists in Taiwan.