國立交通大學

資訊工程系

碩 士 論 文

於 UMTS 行動網路上的安全可證之金鑰委任協定

Provable Secure Scheme with Reliable Key Delegation in UMTS Mobile Networks

研 究 生：沈之涯

指導教授：謝續平　教授

中 華 民 國 九 十 六 年 六 月

於 UMTS 行動網路上的安全可證之金鑰委任協定

Provable Secure Scheme with Reliable Key Delegation in UMTS Mobile Networks

研 究 生：沈之涯　　Student：Chih-Ya Shen

指導教授：謝續平　　Advisor：Shiuh-Pyng Shieh

國 立 交 通 大 學

資 訊 工 程 系

碩 士 論 文

A Thesis

Submitted to Department of Computer Science

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer and Information Science

June 2007

Hsinchu, Taiwan, Republic of China

中華民國 九十六 年 六 月

於 UMTS 行動網路上的安全可證之金鑰委任協定

學生：沈之涯　　指導教授：謝續平 教授

國立交通大學資訊工程學系

摘　　要

於本論文中，我們提出了一個名為 S-AKA 的協定，此協定在解決兩個 UMTS 認證與金鑰交換協定(UMTS AKA)上的安全性問題的同時，亦大幅提升其效能。 UMTS 解決了許多其前一代系統，GSM 系統上的安全性問題，然而，在近期的研究中指出，UMTS 系統上至少還存在兩個安全性上的嚴重問題。這兩個問題就是重導攻擊(Redirection Attack)與中間人攻擊(Man-in-the-middle Attack)。 攻擊者可藉由這兩個攻擊來竊取資料和導致計費上的問題。 同時，原本 UMTS 認證與金鑰交換協定的效率亦是一個問題。 我們在這篇論文中提出 S-AKA 協定以解決上述的問題。S-AKA 在解決重導攻擊與中間人攻擊之餘，亦節省了 30%的頻寬與 25%的訊息數量。

Provable Secure Scheme with Reliable Key Delegation in UMTS Mobile Networks

student：Chih-Ya Shen　　Advisors：Dr. Shiuh-Pyng Shieh

Department of Computer Science
National Chiao Tung University

ABSTRACT

In this paper, a new authentication protocol, S-AKA, is proposed to solve two security problems while enhancing the efficiency of the authentication and key exchange protocol for Universal Mobile Telecommunication System (UMTS AKA). The predecessor of UMTS, Global System of Mobile (GSM) has been shown that it is vulnerable to various attacks. Based on the security framework of GSM, UMTS provides substantial enhancements to solving real and perceived vulnerabilities in GSM and other wireless communication systems. However, two security vulnerabilities of UMTS AKA have been recently discovered, that is, redirection attack and man-in-the-middle attack. An adversary can mount these two attacks to eavesdrop the communication or cause billing problems. On the other hand, the efficiency of UMTS AKA is still worth improving. If a mobile station stays within a SGSN for a long time, the transmission overhead of authentication vectors may incur a huge amount of bandwidth consumption. To solve these problems, S-AKA is proposed in this paper which enhances the security and efficiency of UMTS mobile networks. It defeats redirection attack and man-in-the-middle attack while providing better efficiency than UMTS AKA. Our analysis showed that S-AKA reduced 30% of bandwidth consumption and 25% of message numbers compared with conventional schemes. The security proof of S-AKA is also given to show its security strength.

誌　　謝

能完成這篇論文，首先我要感謝我的指導教授，謝續平老師。 老師在這兩年之中，對我們關懷有加，教導了我許多做研究的方法。老師對我來說，是一座光明的燈塔，讓在濁濁塵世間載浮載沉的我，有了前進的方向。 接著我要感謝我的父母家人，沒有他們給我溫暖的支援，我就沒有辦法專心致志的進行我的研究。 我還要感謝實驗室的學長學姐學弟學妹還有助理們，你們在我淚眼迷茫時帶給我歡笑，在我頹廢喪志時給我信心，謝謝你們大家。

接著我要感謝另外一票人，首先我要感謝阿姐，謝謝你有時候會帶東西來給我吃，和常常陪我走去 D 棚，讓這段原本枯燥乏味的路途增添許多歡樂。再來我要感謝馨允，謝謝你在我大四與碩一的時候陪伴過我，也讓我更加認識了自己，你的碩士論文也要加油呀。然後我要感謝元馨，謝謝你陪我度過了碩一的夏天與秋天，讓我碩一的 seminar 變得有趣了不少，今天你已嫁為人婦，我衷心地祝你幸福。我還要感謝采薇，十幾年的朋友了，雖然升上碩班之後我們相約的時間變少了，但你的電話總是在我最需要你的時候打過來，謝謝你。再來要感謝怡如，在那段我在申請博士班、寫論文初稿最辛苦的時候，天天晚上有你在遠方陪伴著我，雖然我們相隔數百公里，可是你總是讓我感覺到很溫暖，而這溫暖，正是我前進的動力。最後，我要大感謝莞瑄，你的出現為我的生活注入了一股新的活力，謝謝你在我最忙碌的碩二時一直陪著我，和我聊天、出去玩、幫我買衣服和拖鞋、幫我整理房間(Orz)，還有逗我開心，謝謝你。當然，我還要感謝玉米、小花、linkp。玉米你總是陪我去看漫畫吃宵夜，而小花和 linkp 就更要感謝了，我們三劍客上山下海玩遍了北臺灣。 有你們，真好，謝謝你們大家。

目　　錄

圖目錄

表目錄

1.Introduction

Mobile telephony has already become an indivisible part of our everyday lives. With the boost of mobile environments, more and more applications are developed and deployed to provide more and more convenience to the human being. Today, the third generation (3G) mobile phones [9] are used widely together with its precursor, Global System for Mobile (GSM) mobile phones [22]. The goal of third-generation mobile systems is to enhance service capabilities, provide worldwide operation, and improve performance. In the security aspect, 3G mobile systems intend to minify the drawbacks of the second-generation (2G) mobile systems. The drawbacks of 2G mobile systems include: 1) only unidirectional authentication is provided, which may cause the false base station attack, 2) triplets can be reused, and 3) weak encryption.

To address the security weaknesses in GSM, the Universal Mobile Telecommunication System (UMTS) [9] has adopted an enhanced authentication and key agreement protocol, called UMTS AKA. UMTS AKA achieves extra security goals, such as mutual authentication between the mobile station (MS) and the serving network (SN), agreement on an integrity key between the MS and the SN, and freshness assurance of the agreed cipher key and integrity key. The security enhancements in UMTS AKA successfully defeated most of the vulnerabilities discovered in GSM systems, and made UMTS a more secure telecommunication system [21].

Nevertheless, UMTS AKA is still vulnerable to some attacks, such as redirection attack [20] and man-in-the-middle attack [8]. With these attacks, the user may be mischarged or even eavesdropped. Furthermore, the bandwidth consumption of UMTS AKA can still be improved.

In this paper, we will state the security vulnerabilities, and bandwidth bottleneck

of current UMTS AKA. Then we propose our scheme to eliminate the vulnerabilities, and to enhance the efficiency. We will also provide the security and efficiency analysis on UMTS AKA and the proposed scheme.

The remaining part of this paper is organized as follows. In chapter 2, we introduce UMTS AKA and describe the security and bandwidth drawbacks. Chapter 3 is the related work. In chapter 4, we propose our scheme, S-AKA. In chapter 5, we give the security analysis and the bandwidth analysis of both S-AKA and UMTS AKA and compare the two protocols. In chapter 6, we formally prove the security of S-AKA. And in chapter 7, we conclude this paper.

2. Overview of UMTS AKA

UMTS AKA features three main design goals. They are 1) mutual authentication between MS and the network, 2) establishment of a cipher key and an integrity key upon successful authentication, and 3) Freshness assurance to the user of the established cipher and integrity keys. With these three features, UMTS AKA is able to defeat various attacks [21]. In this chapter, we give an overview of UMTS AKA, including the protocol, and the vulnerabilities of UMTS AKA.

2.1. Introduction to UMTS AKA protocol

UMTS AKA adopted the authentication procedure of GSM and resolved the security problems discovered in GSM. UMTS AKA provides new and enhanced security features, such as mutual authentication, integrity key between MS and Serving GPRS Support Node (SGSN), and the guarantee of the freshness of integrity key (IK) and cipher key (CK).

Here, we briefly introduce UMTS AKA. The message flows is depicted in Fig. 1 [9]. There are three entities involved in UMTS AKA, namely, the MS, the SGSN, and the Home Location Register/Authentication Center (HLR/AuC). The MS acts on behalf of the user to communicate with the SGSN and the HLR/AuC to authenticate each other, the SGSN represents the serving network which the MS visits, and the HLR/AuC is in the home domain and is in charge of the authentication data management. The MS and the HLR/AuC share a secret key, K, and some cryptographic algorithms. There are 7 cryptographic functions including f1, f1*, f2, f3, f4, f5, f5*. Functions f1 and f1* are message authentication functions used to compute Message Authentication Code (MAC), function f2 is the message authentication function used to compute RES and XRES, function f3, f4, f5, and f5* are key generation functions used to compute CK, IK,

AK (in normal procedures), and AK (in re-synchronization procedures), respectively. Each of the MS and HLR/AuC maintains a sequence number, SQNMS and SQNHN, respectively. The sequence number can be used to oppose against replay attack.

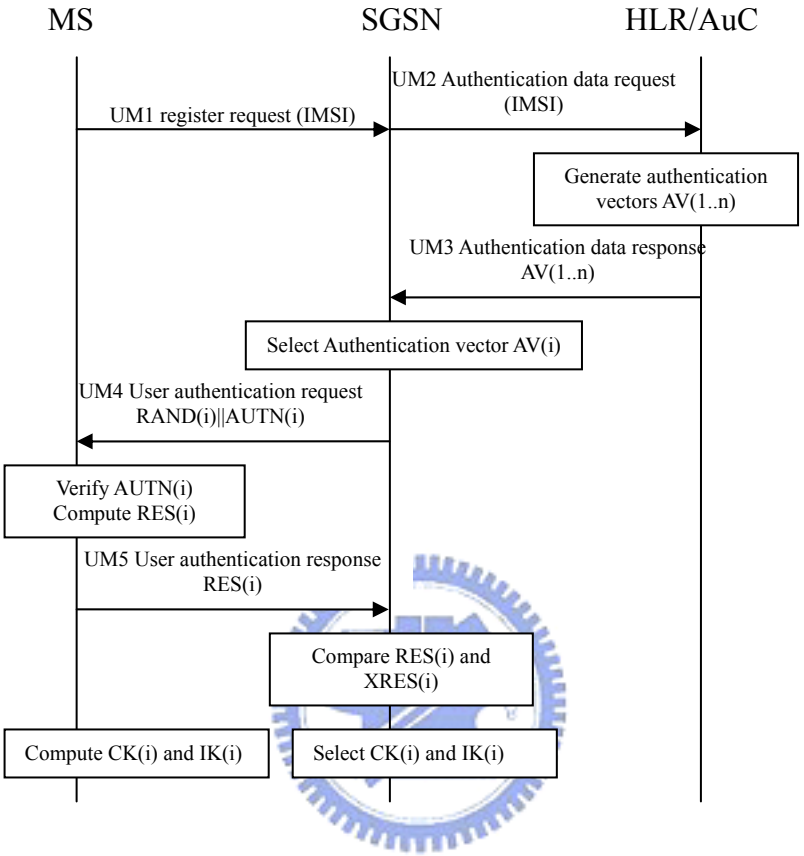


Fig. 1. UMTS AKA

UMTS AKA is shown in Fig. 1,and works as follows.

Step 1 Denoted as UM1. MS sends a registration request containing its International Mobile Subscriber Identity (IMSI) to the Serving GPRS Support Node (SGSN).

Step 2 Denoted as UM2. SGSN passes the request containing IMSI to HLR/AuC.

Step 3 Denoted as UM3. Upon receipt of the request from SGSN, the HLR/AuC sends an ordered array of n authentication vectors to the SGSN. Each authentication vector consists of a random number RAND, an expected response XRES, a cipher key CK, an integrity key IK and an authentication token AUTN.

Step 4 Denoted as UM4. The SGSN selects the next unused authentication vector from the ordered array and sends the parameters RAND and AUTN to the MS.

Step 5 Denoted as UM5. The MS checks whether AUTN can be accepted, and if so, the MS produces a response, RES, which is sent back to the SGSN. The MS also computes the cipher key CK and the integrity key IK. The SGSN compares received RES with XRES. If they match, the authentication procedure completes successfully.

In UMTS AKA mentioned above, the MS authenticates the network at step 5 by checking if the MAC in the AUTN is correct. The MS further verifies if the sequence number in the AUTN is in the correct range. If so, MS successfully authenticates the network. At step 5, MS sends RES to SGSN. The SGSN checks if the RES is correct. If so, SGSN successfully authenticates MS. And thus mutual authentication between MS and SGSN is achieved. Right after MS and SGSN authenticate each other, the cipher key, CK, and the integrity key, IK, are generated for protecting the traffic. The freshness of CK and IK is guaranteed with the sequence number stored in the MS and the SGSN.

2.2. Weaknesses of UMTS AKA

The weaknesses of UMTS AKA can be divided into two categories, namely, security vulnerabilities and efficiency weaknesses. We will describe the security vulnerabilities and efficiency weaknesses in the follows.

2.2.1. Security vulnerabilities

Although UMTS AKA paid much attention on security issues, several security weaknesses are discovered, including redirection attack and man-in-the-middle attack. With these attacks, the adversary can annoy the user with billing problems and can even eavesdrop the communication content. The attacks as described as follows.

Redirection attack

Assume an adversary owns a device which is able to simulate the functionality of a

base station, and at the same time that device can also simulate a normal MS. To the victim MS, the adversary pretends a legitimate base station by broadcasting fabricate base station ID, and to the genuine base station, the adversary pretends to be the victim MS. The scenario is illustrated in Fig. 2.

In Fig. 2, the green dotted line represents the connection between the victim MS and the genuine base station. The victim MS and the genuine base station are both in the home territory. The red solid line is the communication path of the redirection attack.

The adversary can entrap a legitimate user to connect to his base station by broadcasting a bogus base station ID using higher power, and connects to another legitimate foreign network on behalf of the legitimate user. Then, the only thing needed to be done by the adversary is to relay traffic between the legitimate foreign network and the victim without any modification of the communication content. There's one thing worthy of noticing that since the communication content of the victim is protected by the cipher key and integrity key, the adversary cannot modify the content but can only redirect it to another network. The victim will be authenticated by the foreign network because the foreign network is legitimate.

Using this kind of attack, the adversary can persecute the victim with billing problems such as making the victim who is in his home network charged as roaming internationally. Since the foreign network and the genuine foreign base station are both legal, mounting the attack depicted in Fig. 2 can convince the home network that the victim mobile station is in the foreign territory, though the victim mobile station is in the home territory. The home network cannot find out that the victim mobile station is in the home territory since the victim mobile station is not connected with the genuine home base station, and neither can the victim mobile station since the authentication is carried out successfully. Also, the adversary can redirect the victim to a network with weak or no data encryption, such as a false GSM base station mentioned below. Thus the adversary

can eavesdrop the communication content [6].

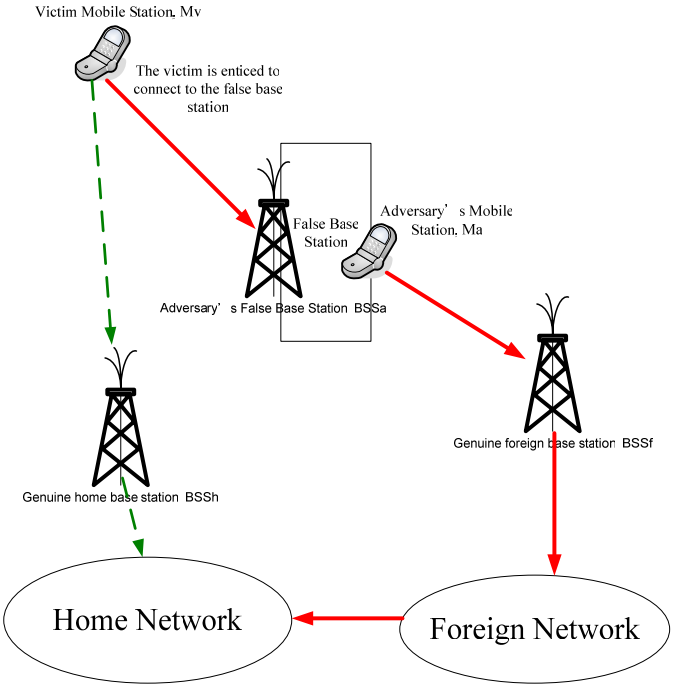


Fig. 2. Redirection attack

Man-in-the-middle attack

Mounting man-in-the-middle attack is able to entrap the victim into using no encryption or weak encryption over the victim's communication and thus an adversary can eavesdrop the whole communication initiated by the victim. The adversary can impersonate a GSM base station and induces the victim 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. The attacker can easily forge GSM cipher mode command and fool the victim into using either no encryption or a weak encryption algorithm. After mounting this man-in-the-middle attack, the attacker can eavesdrop on all mobile station initiated communication since no encryption or weak encryption is applied [8].

The man-in-the-middle attack comprises two phases. The two phases are illustrated in Fig. 3 and Fig. 4, respectively. And the detail of the attack is elaborated as follows.

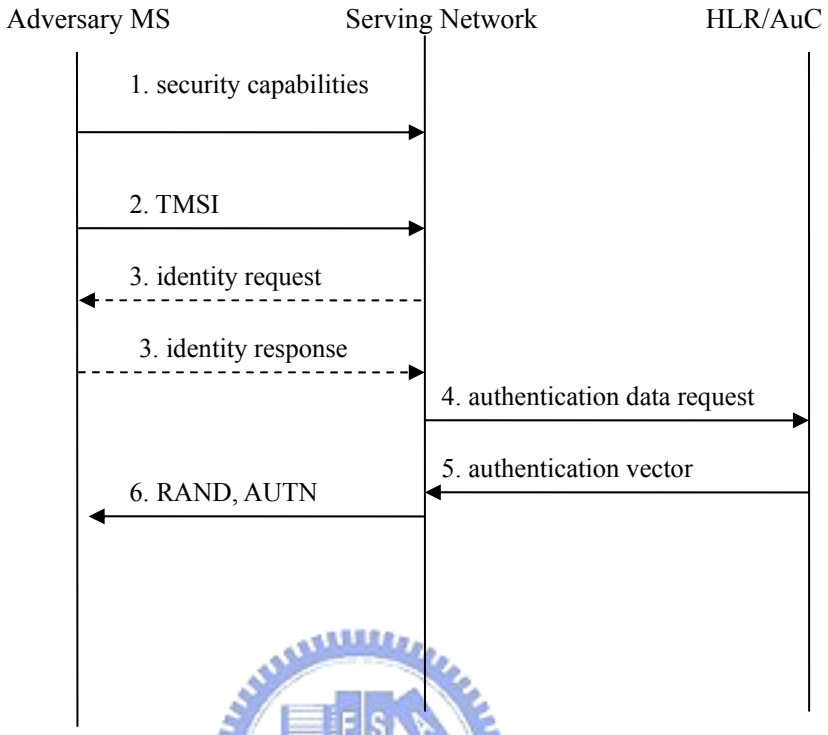


Fig. 3. Man-in-the-middle attack phase 1.

Man-in-the-middle attack, phase 1.

The adversary connects to any legitimate networks on behalf of the victim MS to attain a valid authentication token AUTN. The following steps are carried out:

Step 1 The adversary sends the security capabilities of the victim MS to the serving network during the connection setup.

Step 2 The adversary sends the TMSI of the victim MS to the visited network. If the current TMSI is unknown to the adversary, he sends a faked TMSI (which eventually cannot be resolved by the network).

Step 3 If the TMSI cannot be resolved by the network, the network sends an identity request to the adversary. The adversary replies with the IMSI of the victim.

Step 4 The visited network requests the authentication information for the victim device from its home network.

Step 5 The home network sends the authentication information to the visited network.

Step 6 The network sends RAND and AUTN to the adversary for authentication.

The adversary drops the connection from the visited network.

Since none of the messages sent in steps 1 to 7 are protected by any means, the network cannot recognize the presence of the adversary. Consequently, the attacker obtains an authentication token which he in turn can use in phase 2 of the attack to impersonate a network to the victim device.

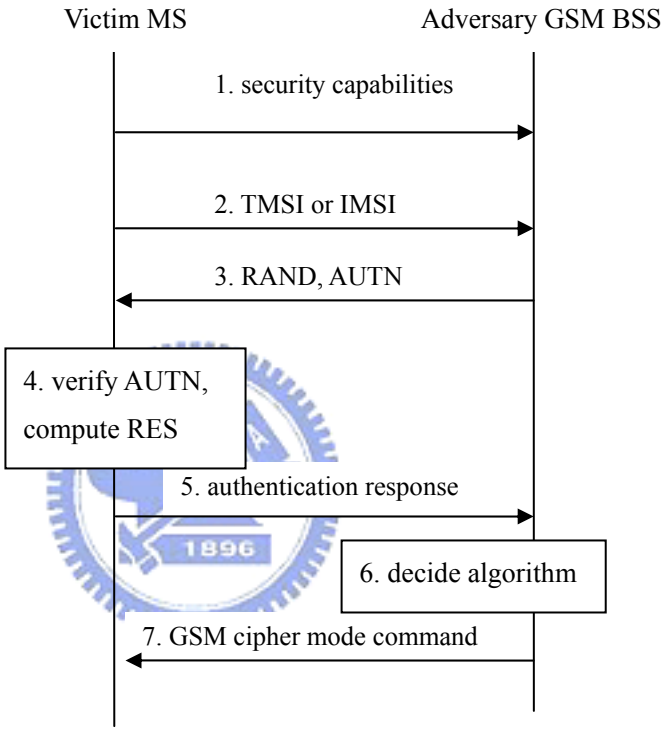


Fig. 4. Man-in-the-middle attack phase 2.

Man-in-the-middle attack, phase 2:

The adversary impersonates a valid GSM base station to the victim MS.

Step 1 The adversary and the victim MS establish a connection and the MS sends its security capabilities to the adversary.

Step 2 The victim MS sends its TMSI or IMSI to the adversary.

Step 3 The adversary sends the victim the authentication challenge RAND and the authentication token AUTN he obtained from the real network in phase 1 of the attack.

Step 4 The victim MS successfully verifies the authentication token and the adversary is considered to be a legitimate network.

Step 5 The victim MS replies with the authentication response.

Step 6 The adversary decides to use “no encryption” The MS accepts the authentication token if the token is fresh, i.e., not too much time has elapsed between phase 1 and phase 2.

Step 7 The adversary sends the MS the GSM cipher mode command including the chosen encryption algorithm.

Note that the adversary does not allow the intruder to impersonate the MS to the network at the same time. In order to allow for a regular use of the connection by the victim unit, the attacker has to establish a regular connection to a real network to forward traffic it receives from the MS. As a side effect the attacker has to pay the cost for this connection.

2.2.2. Efficiency weaknesses of UMTS AKA

When UMTS AKA is being performed, after the SGSN sends HLR/AuC the authentication data request, the HLR/AuC replies the SGSN with n authentication vectors (AV). If the MS stays within the same SGSN for a long time and the n AV are exhausted, the SGSN must again send HLR/AuC the authentication data request for another n AV. The transmission of authentication data request and AV consumes a huge amount of bandwidth, and the authentication data request may be expensive since the SGSN and the HLR/AuC may be located in different countries. Furthermore, the number of AV sent by the HLR/AuC to the SGSN is also important. If the MS stays in the same SGSN for a long time, a small n value will consume much more bandwidth than a larger n. However, since it is difficult to anticipate how long the MS will stay in the same SGSN, it is also difficult to choose an appropriate n value. Fig. 5 shows the bandwidth

consumption of UMTS AKA on different n values. As shown in the figure, the bandwidth consumption of UMTS AKA when n = 2 is the smallest in the beginning, but is the largest when the number of registrations is more than 100. Conversely, the bandwidth consumption of UMTS AKA where n= 50 is the largest initially, but it becomes the smallest when the number of registration reaches 400. Since we don't know how many times of registration will the MS perform, we can not choose the most appropriate n value to achieve the best bandwidth efficiency.

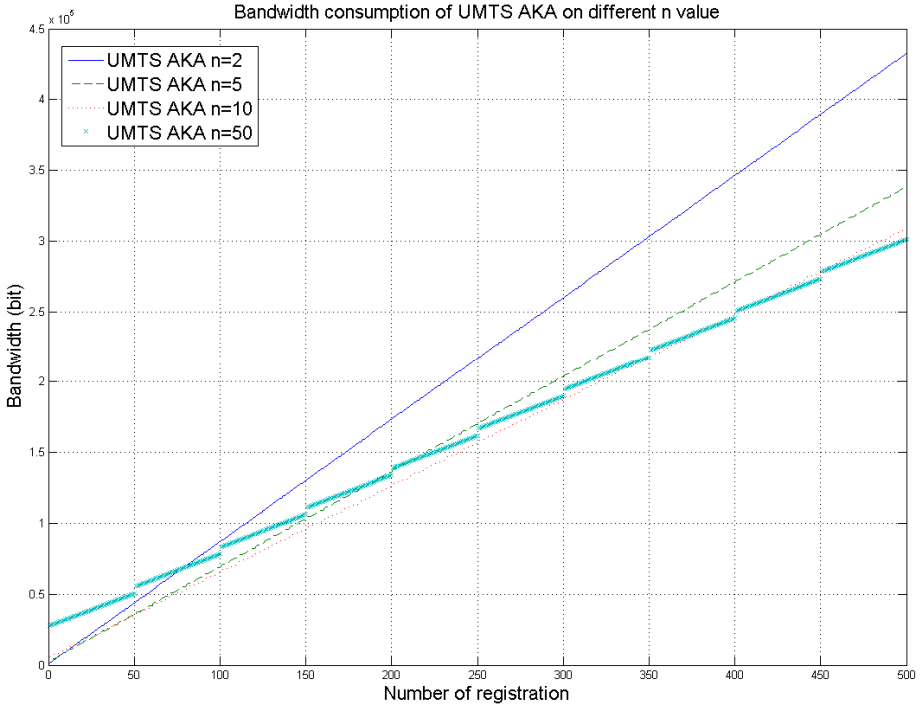


Fig. 5. Bandwidth consumption of UMTS AKA on different n value

3. Related work

There are some recently published papers which aim to enhance the security of UMTS AKA and decrease the bandwidth consumption. Some of them use asymmetric-key protocols [1][3], and the others use symmetric-key protocols [2][4][5][6][7]. The main shortcomings of asymmetric-key protocols are that they require the construction of large-scaled (or even, global) public key infrastructure in order to support global roaming and they require more processing power of mobile station and the bandwidth consumption in exchanging public keys. Employing asymmetric-key protocols may require a large amount of modification on current UMTS architecture, including the modification on the core-network, mobile station and the USIM card. These shortcomings make it very difficult to use asymmetric-key protocols for authentication in UMTS. On the other hand, since there is already a key pre-shared in both the home network and the user's USIM card, adopting symmetric-key protocols seems much more economically efficient compared to asymmetric-key protocols. Also, using symmetric-key protocols needs much less computational power than asymmetric-key protocols do.

M. Zhang et al. proposed an enhancement on UMTS authentication and key agreement protocol [6], which can solve the problem of redirection attack. They proposed a new protocol, "AP-AKA," to deal with the redirection attack. Nevertheless, the proposed protocol, "AP-AKA," can't solve the false base station attack and the bandwidth consumption is higher than the original UMTS AKA.

C. -M. Huang et al. proposed an authentication protocol which pruned off the authentication vector transmission in the UMTS AKA, achieve bilateral authentication between MS and SN, and reduce the stored space in SN [5]. Unfortunately, they didn't

take false base station attack and redirection attack into consideration, which means that the proposed protocol may be prone to eavesdropping and billing problem.

J. Al-Saraireh et al. proposed a new authentication protocol to improve the efficiency [2]. In their scheme, the MS is responsible for generating the authentication vector, which was the responsibility of the HLR in UMTS AKA. The scheme proposed decreased the time delay, call setup time, and signaling traffic. However, it relies on the MS's computation power to generate the authentication vectors and it also suffers from redirection attack as well as man in the middle attack.

4. Proposed scheme S-AKA

Before we elaborate our proposed scheme, we first state our assumption of the environment. We have the following assumptions, 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 [9], and 3) the user trusts the HLR/AuC [9]. The design goals of our proposed scheme includes the follows, 1)defeat redirection attack, 2) defeat man-in-the-middle attack, 3) mutual authentication between MS and HLR/AuC, 4) mutual authentication between MS and SGSN, 5) establishment of a cipher key and an integrity key upon successful authentication, 6) freshness assurance to the user of the established cipher and integrity keys, and 7) reduce the bandwidth consumption. With these goals, our proposed scheme can be said to be secure and efficient.

For conciseness, when we describe our proposed scheme, we will use abbreviations. The abbreviations are listed in Table 2. Also, there are some symbols we will use throughout this chapter, and they are listed in Table 1.

Table .1 Symbols.

||	Concatenation
f1	Message authentication function used to compute MAC
f2	Message authentication function used to compute RES and XRES
f3	Key generating function used to compute CK
f4	Key generating function used to compute IK
f5	Key generating function used to compute AK
f6	Key generation function used to compute DK
f7	Key generation function used to compute PLK
K	Long-tem secret key shared between the USIM and the AuC

Table .2 Abbreviations

AK	Anonymity Key
AKA	Authentication and key agreement

AMF	Authentication management field
AUTN	Authentication Token
CK	Cipher Key
DK	Delegation Key
FRESH	A counter of the number of authentications
IK	Integrity Key
IMSI	International Mobile Subscriber Identity
LAI	Location Area Identity
MAC	The message authentication code generated by f1
MS	Mobile Station
PLK	Payload Encryption Key
RAND	Random challenge
XRES	Expected Response

4.1. Proposed scheme

To meet the design goals mentioned above, we proposed a new authentication scheme, S-AKA.

To solve redirection attack, S-AKA needs the assistance of the MS itself and the SGSN. The MS is responsible for rejecting illegal base station connection, and the SGSN is responsible for verifying the Location Area Identifier (LAI) sent from the MS. If the LAI is illegal , the SGSN will drop the connection. The LAI in UMTS AKA was not encrypted by any means, and thus can be altered by the adversary in order to successfully mount the redirection attack. In S-AKA, we use Message Authentication Code (MAC) to protect the integrity of the LAI. If someone tries to modify the LAI, the illegal modification will be detected immediately.

To solve the man-in-the-middle attack, S-AKA introduces another key, PLK, to encrypt the payload. In case of connecting to a GSM base station, the MS and the SGSN will generate the PLK to encrypt and decrypt the messages between them. PLK makes the adversary not able to eavesdrop and modify the communication. Since in UMTS AKA, there is no mechanism to generate the PLK, we introduce a new key generation

function, f7, to generate PLK.

Our scheme uses a ticket based authentication scheme for bandwidth reduction [5][10]. The ticked based authentication scheme make the HLR/AuC authorize the SGSN for subsequent mutual authentication between SGSN and MS. After the first time the HLR/AuC authenticates the MS, the HLR/AuC sneds the delegation key (DK) to SGSN. The SGSN then uses the DK for successive 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. There is no DK generation function in UMTS AKA, so we use a new key generation function, f6, to generate DK.

S-AKA can be divided into two parts. The first part, called S-AKA-I is the authentication procedure which takes place when it is the first time the MS and the SGSN authenticate each other, and the second part, S-AKA-II, is the authentication procedure takes place when it is more than the second time the MS and the SGSN want to authenticate each other. In the first part, S-AKA-I, the SGSN would communicate with the HSS/AuC to obtain authorization and delegation for proceeding S-AKA-II. In S-AKA-II, the MS and the SGSN can authenticate each other without the data transmission between SGSN and HSS/AuC, and this may reduce the bandwidth consumed by the authentication procedure.

The proposed S-AKA-I and S-AKA-II are illustrated in Fig. 4 and Fig. 5, respectively. The messages of S-AKA are explained as follows.

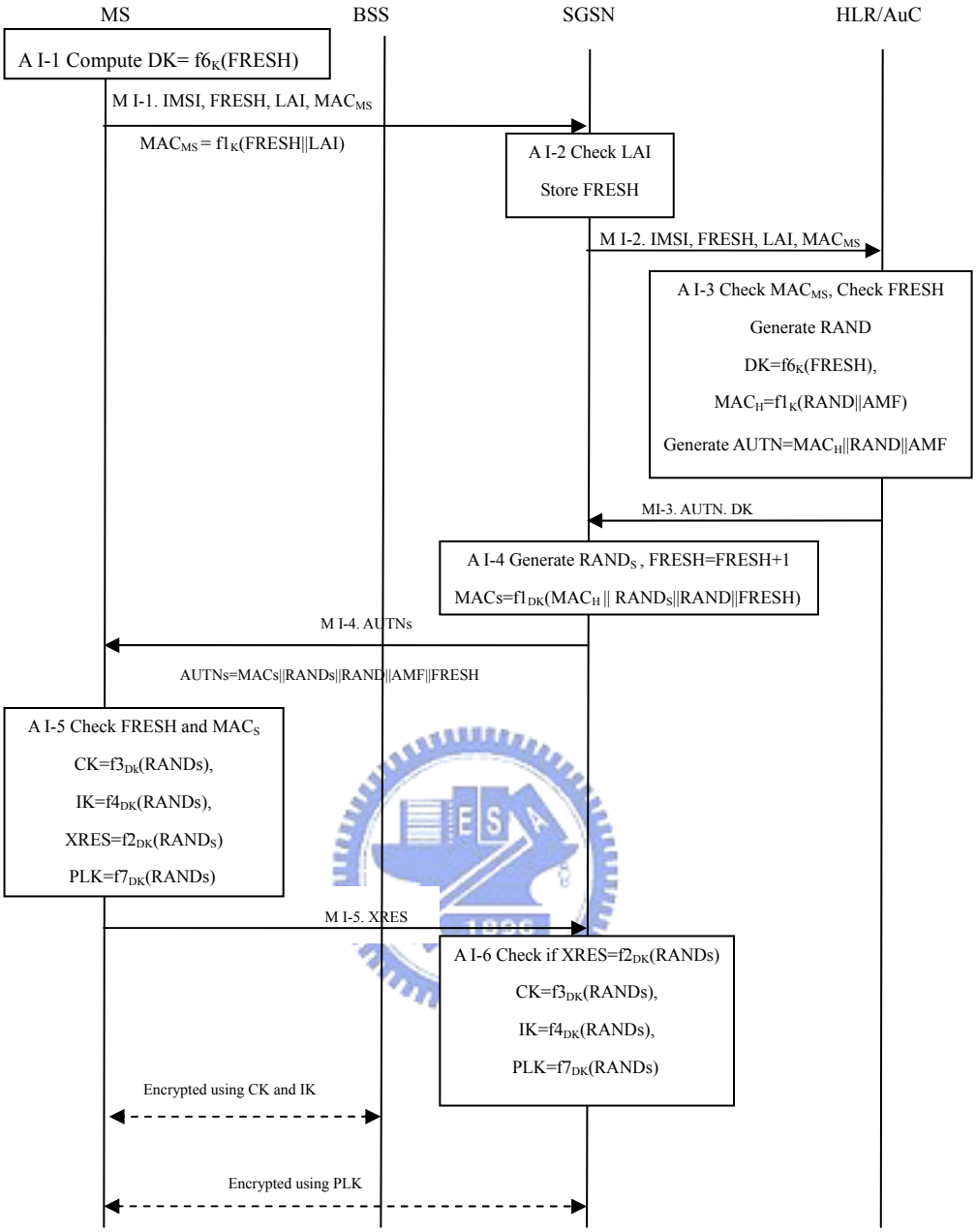


Fig. 6. The first part of S-AKA, S-AKA-I

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

Before sending the request to SGSN, the MS computes the delegation key, DK, as *$DK = f6_K(FRESH)$,* where K is the pre-shared secret key stored in both USIM and HLR/AuC. This key will be used on generating MAC_{MS} in M I-1.

MS sends a registration request to the SGSN through base station BSS. This message comprised four parts, namely, IMSI, FRESH, LAI, and MAC_{MS}. IMSI is the identity of user. FRESH is the number of authentications and will be accumulated by

one when one authentication completes successfully. FRESH is also used for generating DK when it is the first time MS connects to this SGSN. LAI is the location area identifier of the base station with which the MS connects, and is used to defeat the redirection attack.. The MAC_{MS} is the message authentication code which can be computed as $MAC_{MS} = f1_K(FRESH||LAI)$, where f1 is the message authentication code function, DK is delegation key generated above. The MACMS is used to protect the integrity of the tokens, FRESH and LAI.

Step 2. SGSN sends IMSI, FRESH, LAI, MAC_{MS} to HLR/AuC

Right after receiving the registration request, the SGSN checks LAI to see if the base station is physically connected to the SGSN. If SGSN finds that the base station is not connected to itself, the SGSN rejects the request immediately. The SGSN stores the FRESH and then the SGSN passes IMSI, FRESH, LAI, and MAC_{MS} to HLR/AuC.

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

Upon receipt of the message, HLR/AuC first verifies if MAC_{MS} equals $f1_K(FRESH||LAI)$. If not, FRESH or LAI may have been modified and HLR/AuC rejects the request. HLR/AuC verifies if the FRESH is smaller than it should be, if so, the registration request will be rejected since this may be a replayed message. To delegate SGSN to authenticate the MS, HLR/AuC generates a delegation key, DK, and other authentication parameters to verify the legality of the MS. HLR/AuC generates a random number RAND and computes a delegation key DK as $f6_K(FRESH)$. HLR/AuC then computes a message authentication code, MAC_H, as $f1_K(RAND||AMF)$, where AMF is the authentication management field [9]. Finally, HLR/AuC generates AUTN as $(MAC_H||RAND||AMF)$. HLR/AuC sends DK and AUTN to the SGSN. After this message, HLR/AuC has successfully delegated SGSN to authenticate the MS.

Step 4. SGSN sends $AUTN_S$ to MS

Upon receiving DK and AUTN from HLR/AuC, SGSN stores DK and AUTN. Then, SGSN increases FRESH by one. Afterward, SGSN generates a random number $RAND_S$ and computes the message authentication code MAC_S as $MAC_S = f1_{DK}(MAC_H || RAND_S||RAND||FRESH)$, where MAC_H, RAND and DK are received from HLR/AuC. Finally, SGSN constructs $AUTN_S$, where $AUTN_S = MAC_S || RAND_S || RAND || AMF. ||$ FRESH and "||" denotes concatenation. SGSN then sends the $AUTN_S$ to MS, where the $AUTN_S$ consists MAC_S, $RAND_S$, RAND, AMF, and FRESH.

Step 5. MS sends XRES to SGSN

The MS authenticates the SGSN by verifying MAC_S. The MS first checks FRESH to see if it is larger than MS's n. If so, MS set its FRESH to the one received in $AUTN_S$. If not, MS rejects. Then, MS computes $XMAC_H = f1_K(RAND || AMF)$ where RAND and AMF are from the received $AUTN_S$. The MS also computes $XMAC_S = f1_{DK}(XMAC_H || RAND_S ||RAND||FRESH)$. Where RAND and $RAND_S$ are retrieved from $AUTN_S$ and FRESH is the times of performing authentication procedures. Then MS checks if the following equation holds.

$$XMAC_S = MAC_S.$$

If the equation holds, it means that the SGSN is authenticated by the MS. If the equation doesn't hold, it means at least one of HLR/AuC or SGSN is invalid. MS will reject. If both HLR/AuC and SGSN are valid, the MS computes the expected response message as $XRES = f2_{DK}(RAND_S)$. MS then computes an integrity key, IK, where IK = f4DK(RANDS), and a cipher key, CK, where CK = f3DK(RANDS). Then, MS checks if the base station it connects with is a GSM base station or an UMTS base station. If the base station is a GSM base station, an extra payload encrypt key, PLK, will be generated to protect the payload between the MS and the SGSN. PLK is used to encrypt

the data before CK and IK. The encrypted data will be decrypted on SGSN and thus protects the data confidentiality against false GSM base station attack. MS sends XRES to SGSN, where XRES is computed as XRES = f2DK(RANDS).

After receiving XRES from the MS, the SGSN checks if XRES = $f1_{DK}(RAND_S)$. If yes, it means the MS is legitimate. And the SGSN computes an integrity key IK as IK = $f4_{DK}(RAND_S)$, and a cipher key CK as CK = $f3_{DK}(RAND_S)$. SGSN subsequently checks if the base station MS connects with is a GSM base station or an UMTS base station. If the base station is a GSM base station, the PLK will also be computed to decrypt the data transmitted from the MS. Finally, the SGSN accumulates FRESH by one for indicating the number of successful authentications.

The following is the second part of the S-AKA protocol, S-AKA-II. When the MS connects to the same SGSN for several times, the S-AKA-II will be executed in order to decrease the bandwidth consumption. S-AKA-II is illustrated in Fig. 5. And the message flow is explained in detail as follows.

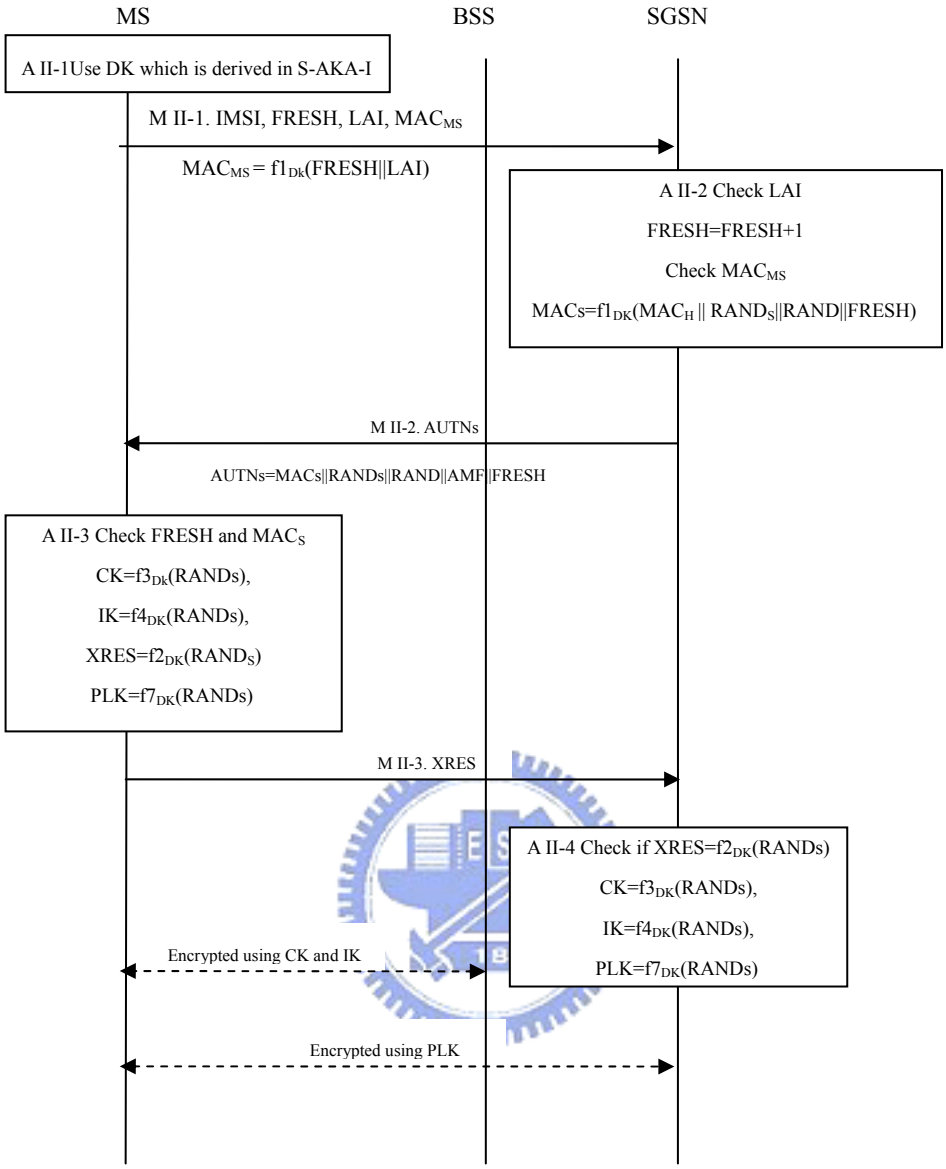


Fig. 7. The second part of S-AKA, S-AKA-II.

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

The MS then utilize the delegation key derived in A I-1, S-AKA-I for upcoming authentications in the same SGSN. Then, MS sends a registration request to then SGSN through base station BSS. This message is similar to M I-1 in S-AKA-I.

Step 2. SGSN sends the $AUTN_S$ to MS

In S-AKA-II, the SGSN already has the following parameters, namely, FRESH, $RAND_S$, DK, AMF, MAC_H, n, and RAND. They were obtained in S-AKA-I. These

parameters will help the SGSN and the MS authentication each other without the aid of the HLR/AuC. The SGSN first checks LAI to see if the base station is physically connected to the SGSN. If SGSN finds that the base station is not connected to itself, the SGSN rejects the request immediately. The SGSN then verifies FRESH received from Message II-1 to see if it is a replayed message. If not, the SGSN checks the MAC_{MS} on behalf of the HLR. If the SGSN finds the MAC_{MS} is not legitimate, the SGSN rejects the connection. The SGSN computes the message authentication code MAC_S as $MAC_S = f1_{DK}(MAC_H \| RAND_S \| RAND \| FRESH)$, where MAC_H, RAND and DK are received from HN. SGSN then constructs $AUTN_S$, where $AUTN_S = MAC_S \| RAND_S \| RAND \| AMF. \| FRESH$. Finally, the SGSN sends the $AUTN_S$ to MS, where the $AUTN_S$ consists of MAC_S, $RAND_S$, RAND, AMF, and FRESH.

Step 3. The MS sends XRES to SGSN

MS authenticates the SGSN and the HLR by verifying MAC_S and MACH, respectively. This step is similar to step A I-5 in S-AKA-I. Then, MS sends XRES to SGSN, where XRES is computed as XRES = f2DK(RANDS). The SGSN verifies the legitimacy of the MS. This step is similar to A I-6 in S-AKA-I.

5. Security and efficiency analysis of S-AKA

In this chapter, we will examine S-AKA in two aspects, namely, security and efficiency. In the security analysis, we analyze the security of S-AKA against various attacks. On the other hand, in the efficiency analysis, we analyze the number of messages and the bandwidth consumption of S-AKA and compare S-AKA with UMTS AKA.

5.1. Security analysis

In this section, we elaborate how S-AKA defeat various attacks listed in the design goal of S-AKA. 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 in the proposed S-AKA protocol.

5.1.1. Security against redirection attack

For covering all circumstances, we divide the scenario into two cases according to the behavior of the adversary's base station. One is the adversary's base station broadcasts a foreign base station LAI to pretend it's in the foreign territory, and the other is the adversary's base station broadcasts a local base station LAI to pretend it's in the victim MS's home territory. We describe how S-AKA can defeat both of these two cases as follows.

Case 1. The adversary's base station pretends to be in the foreign territory.

Assume the adversary's base station broadcasts the LAI which is in the foreign territory. Since the MS can monitor the status of the base stations nearby, the MS will

first choose to connect to those base stations belonged to the home territory. Thus the MS will not connect to the adversary's base station unless the adversary's base station jam the whole spectrum and broadcast its LAI with higher power to convince the MS that there is no other base stations except the adversary's. However, the user will still discover that he connects to a foreign network since the foreign network ID will be shown on the MS.

Case 2. The adversary's base station pretends to be in the home territory.

In this case, the MS is not able to distinguish the genuine base station from the adversary's since they all are in the home territory. The adversary's base station broadcasts its LAI using higher power and thus can entice the MS to connect with him. However, the SGSN in the foreign network can help the MS out. In message 1 of S-AKA depicted in Fig. 4, the MS will send the base station LAI to the SGSN. Upon receipt of the LAI of the base station, the SGSN will first check if the LAI of the base station is indeed physically connected the SGSN. If not, the SGSN will reject the connection immediately. Thus, if the adversary's base station pretends to be in the home network and intended to redirect the connection to a foreign network, the connection will be dropped by the SGSN when the SGSN finds out that the adversary's base station is not in the SGSN's territory.

In the two cases mentioned above, we described that the redirection attack cannot be carried out when S-AKA is used. This not only helps user from suffering billing problems but also helps them out from being redirected to a network with weak encryption key. In the following section, we describe how S-AKA defeat man in the middle attack.

5.1.2. Security against man in the middle attack

To defeat the man-in-the-middle attack, we introduce a payload encrypt key, PLK.

When MS finds out that the base station it's connecting to is a GSM base station, it will compute the PLK right after receiving M I-4 of S-AKA-I and M II-2 of S-AKA-II in Fig. 6 and Fig. 7, respectively. The MS then encrypt the data using the PLK to provide data confidentiality between the MS and the SGSN. Even if the adversary's false GSM base station chose not to encrypt the data, the PLK will still protect the data confidentiality.

The SGSN will also compute the PLK right after receiving the M I-5 of S-AKA-I and M II-3 in Fig. 6 and Fig. 7 to decrypt the data encrypted with PLK by MS when the SGSN finds out the data is received from a GSM base station. Since the encryption process with PLK involved may be implemented using simple exclusive-OR operations, the encrypt/decrypt operations will not consume too much computation power and thus the efficiency and the data confidentiality will still remain.

5.1.3. Mutual authentication between MS and HN

HN authenticates MS by message M I-2. HLR/AuC checks the parameters FRESH and MAC_{MS}.

MS authenticates HN when receiving the AUTNs from the SGSN (message M I-4 and M II-2). The AUTNs includes MACs, RANDs, RAND, AMF, and n. MS can compute the MAC of HN, $XMAC_H$ using the parameters RAND and AMF. However, since MAC_H is not included in $AUTN_S$, MS has no way to verify if the $XMAC_H$ and MAC_H are the same. Therefore, MS authenticates HN by computing the $XMAC_S$ as $XMAC_S = f1_{DK}(XMAC_H \,||\, RAND_S \,||RAND||FRESH)$. The equation above holds only if the HN and the SN are both valid.

5.1.4. Mutual authentication between MS and SGSN

The SGSN authenticates the MS by verifying the XRES in message M I-5 and M

II-3. If XRES equals $f2_{DK}(RAND_S)$, the MS is authenticated.

On the other hand, when the MS intends to authenticate the SGSN, it first computes the $XMAC_H$ as $XMAC_H = f1_K(RAND \,||\, AMF)$, where RAND and AMF are from the received $AUTN_S$. And then the MS computes the $XMAC_S$ as $XMAC_S = f1_{DK}(XMAC_H \,||\, RAND_S \,||RAND||FRESH)$, where RANDS, n, and RAND are obtained from AUTNS. The MS then verifies if $XMAC_S$ equals MAC_S. If so, the SGSN is successfully authenticated.

5.1.5. Cipher key and integrity key establishment and freshness assurance

In S-AKA, the ciper key and the integrity key are negotiated in A I-5 and A I-6 in S-AKA-I. And in S-AKA-II, the cipher key and the integrity key are negotiated in A II-3 and A II-4. The freshness of the cipher key and integrity key are guaranteed by the counter FRESH in the AUTNs.

5.1.6. Security Against Replay Attack

In S-AKA, the adversary can replay M I-1, M I-4, and M I-5 in S-AKA-1 or M II-1, M II-2, and M II-3 in S-AKA-2. Since M I-1 is In the following, we examine the six messages which would be replayed and explain the difficulties of replaying the three messages.

1. **M I-1. IMSI, FRESH, LAI, MAC_{MS}**

   Since the parameter FRESH is the number of authentications taken by the MS and FRESH is protected by MAC_{MS}, a replayed message will be discovered immediately and the connection will be dropped.

   The security against replay attack of M II-1 in S-AKA-II is similar to those mentioned above.

2. **M I-4. AUTNs**

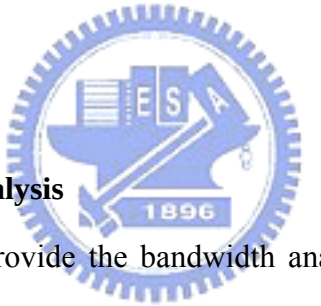
AUTNs contains MAC_S, $RAND_S$, RAND, AMF, and n, where n is the number of authentications on that SGSN. If a message is replayed, MS will discover the attack and drop the connection.

The security against replay attack of M II-2 in S-AKA-II is similar to those mentioned above.

3. **M I-5. XRES**

XRES is computes as $XRES_S = f2_{DK}(RAND_S)$. Since RANDS changes every time the authentication is performed, replayed XRES will not be accepted by SGSN.

The security against replay attack of M II-3 in S-AKA-II is similar to those mentioned above.



5.2. Bandwidth analysis

In this subsection, we provide the bandwidth analysis on both UMTS AKA and our S-AKA and compare the two protocols. In our environment, we assume that in UMTS AKA, the HLR/AuC sends back a batch of *m* authentication vectors each time, and the MS and the SGSN authenticate each other for *p* times. With these two assumptions, we can compare UMTS AKA and S-AKA fairly. In the following sections, we first compute the bandwidth consumption and message number of UMTS AKA, then we compute the bandwidth consumption and message number of S-AKA. And finally we compare these two protocols.

5.2.1. Performance Analysis of UMTS AKA

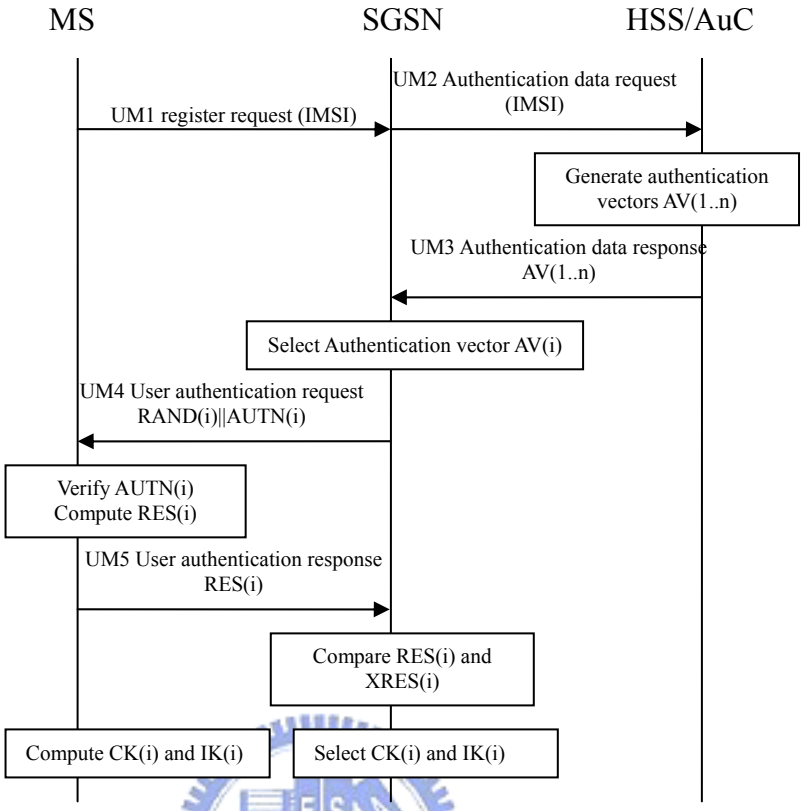


Fig. 8. UMTS AKA

Fig. 8 shows the UMTS AKA. The messages between MS, SGSN, and HLR/AuC are UM1 through UM5. The size of these five messages are calculated as follows.

UM1 is the first message which is comprised of three parameters, namely, IMSI, Service Request, and LAI. The length of UM1, denoted as L(UM1) is calculated as

$$L(UM1) = L(IMSI) + L(Service\ Request) + L(LAI)$$
$$= 128 + 8 + 40 = 176\ bits \qquad (4.1)$$

UM2 is the second message which contains the same parameters as UM1. Thus

$$L(UM2) = L(UM1) = 176\ bits \qquad (4.2)$$

UM3 contains a batch of authentication vectors(AV). The length of each AV is calculated as

$$L(AV) = L(RAND) + L(XRES) + L(CK) + L(IK) + L(AUTN)$$
$$= 128 + 32 + 128 + 128 + 128 = 544\ bits \qquad (4.3)$$

In our assumption, we assumed that each time the HLR/AuC returns a batch of m AVs.

Therefore the length of UM3 is calculated as

$$L(UM3) = m * L(AV) = m * 544 \text{ bits} \qquad (4.4)$$

UM4 is comprised of the parameters RAND and AUTN, where AUTN = (SQN $\oplus$ AK || AMF || MAC) and the length of AUTN is computed as follows.

$$L(AUTN) = MAX(L(SQN), L(AK)) + L(AMF) + L(MAC)$$
$$= 48 + 16 + 128 \text{ bits}$$

The length of UM4 is computed as

$$L(UM4) = L(RAND) + L(AUTN) = 128 + 128 = 256 \text{ bits} \qquad (4.5)$$

UM5 only contains the RES.

$$L(UM5) = L(RES) = 32 \text{ bits} \qquad (4.6)$$

Two cases listed below may consume different bandwidth.

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

$$L(UM1) + L(UM2) + L(UM3) + L(UM4) + L(UM5)$$
$$= 176 + 176 + m*544 + 256 + 32 = 640 + m*544 \text{ bits} \qquad (4.7)$$

Case 2. If the SGSN has unused AVs, only UM1, UM4 and UM5 must be transmitted. Therefore the bandwidth consumption is

$$L(UM1) + L(UM4) + L(UM5) = 176 + 256 + 32 = 464 \text{ bits} \qquad (4.8)$$

As our assumption, we assumed that the MS and SGSN authenticate each other for p times, hence the overall bandwidth consumption is calculated as follows.

Bandwidth Consumption of UMTS AKA

The bandwidth consumption is given as

$$\begin{cases} \left\lceil \frac{p}{m} \right\rceil \bullet \left[(640 + 544 \bullet m)\right] + \left[(p\%m) - 1\right] \bullet 464 \quad bits, if \quad p\%m \neq 0 & (4.9) \\ \frac{p}{m} \bullet \left[(640 + 544 \bullet m)\right] + (m-1) \bullet 464 \quad , if \ p\%m = 0 & (4.10) \end{cases}$$

The number of messages is given as

$$\left\lceil \frac{p}{m} \right\rceil \bullet 5 + (p-1) \bullet 3 \qquad (4.11)$$

5.2.2. Performance Analysis of S-AKA

The messages of the two parts of S-AKA are illustrated in Fig. 4 and Fig. 5, respectively. We first analyze the bandwidth consumption of S-AKA-I, and then S-AKA-II. The analysis is shown as follows.

M I-1 is the first message which is comprised of five parameters, namely, IMSI, Service Request, LAI, FRESH, and MAC_{MS}. The length of M I-1 is calculated as

$$L(M\ I\text{-}1) = L(IMSI) + L(Service\ Request) + L(LAI) + L(FRESH) + L(MAC_{MS}) = 128 + 8 + 40 + 24 + 64 = 264\ bits \quad (4.12)$$

M I-2 is the second message which contains the same parameters as M I-1. Thus $L(M\ I\text{-}2) = L(M\ I\text{-}1) = 264\ bits$ (4.13)

M I-3 contains AUTN and DK, where AUTN is comprised of MAC_H, RAND, and AMF. The length of AUTN is calculated as

$$L(AUTN) = L(MAC_H) + L(RAND) + L(AMF) = 64 + 128 + 16 = 208\ bits \quad (4.14)$$

And the length of this message is computed as

$$L(M\ I\text{-}3) = L(AUTN) + L(DK) = 208 + 128 = 336\ bits \quad (4.15)$$

M I-4 contains AUTN, where AUTN = (MAC_S||$RAND_S$ || RAND||AMF||FRESH) and the length of AUTN is

$$L(AUTN) = L(MAC_S) + L(RAND_S) + L(RAND) + L(AMF) + L(FRESH) = 64 + 128 + 128 + 16 + 24 = 360\ bits \quad (4.16)$$

M I-5 only contains the XRES. L(M I-5) = L(XRES) = 32 bits (4.17)

We now analyze the second part of S-AKA, S-AKA-II, as follows.

M II-1 is identical to M I-1, hence the length of L(M II-1) is

$$L(M\ II\text{-}1) = L(M\ I\text{-}1) = 264\ bits \quad (4.18)$$

M II-2 is exactly the same as M I-4, and the length of M II-2 is

$$L(M\ II\text{-}2) = L(M\ I\text{-}4) = 360\ bits \quad (4.19)$$

M II-3 is also identical to M I-5. Thus the length of this message is

$$L(M\ II\text{-}3) = L(M\ I\text{-}5) = 32\ bits \quad (4.20)$$

There are two cases which may consume different bandwidth, and are listed as follows.

Case 1. If it is the first the MS meets the SGSN, the S-AKA-I must be performed. The bandwidth consumption is

$$L(M\ I\text{-}1) + L(M\ I\text{-}2) + L(M\ I\text{-}3) + L(M\ I\text{-}4) + L(M\ I\text{-}5) = 264 + 264 + 336 + 360 + 32 = 1256\ bits \quad (4.21)$$

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

$$L(M\ II\text{-}1) + L(M\ II\text{-}2) + L(M\ II\text{-}3) = 264 + 360 + 32 = 656\ bits \quad (4.22)$$

As our assumption, we assumed that the MS and SGSN authenticate each other for p times, hence the overall bandwidth consumption is calculated as follows.

Bandwidth consumption of S-AKA

$$\begin{cases} 1256 + (p-1) * 656 \quad bits, p \geq 1 \\ 0\, bits, otherwise \end{cases} \quad (4.23)$$

The number of messages is given as

$$2 + p \cdot 3 \quad (4.24)$$

5.2.3. Comparison

In this subsection, we show the comparisons of S-AKA and UMTS AKA.

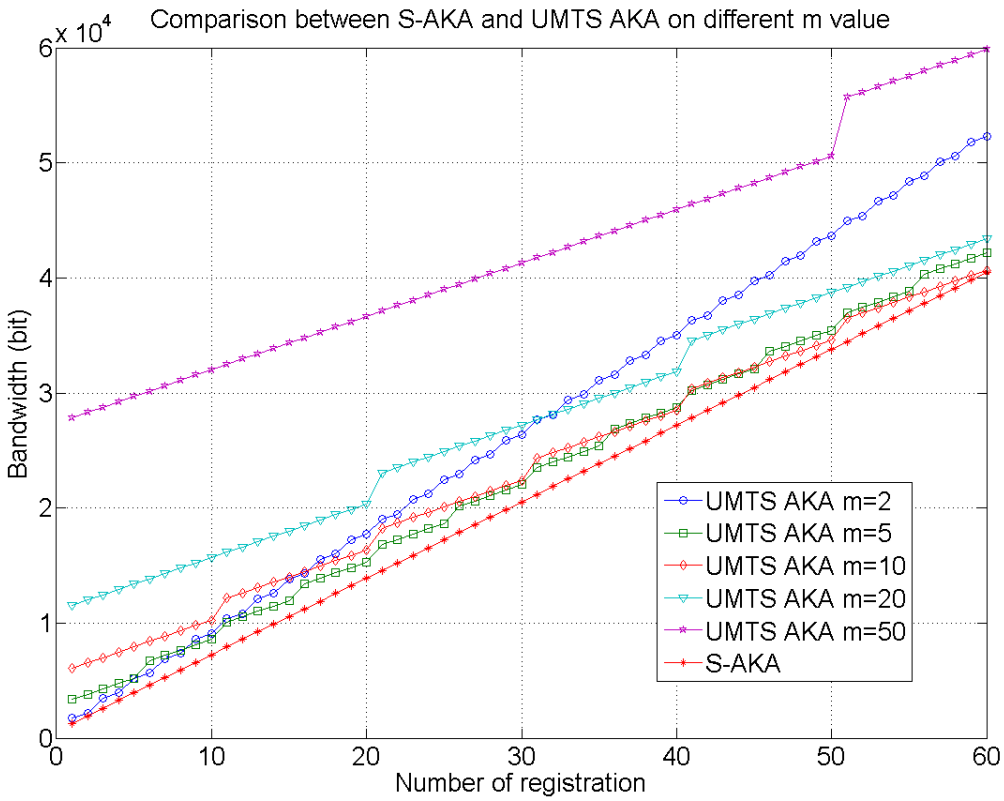


Fig. 9. The comparison between S-AKA and UMTS AKA on different m values

Fig. 9 illustrates the bandwidth consumption comparison between S-AKA and UMTS AKA on different m value. The value, m, is the number of transmitted authentication vectors from HLR to SGSN in UM3 in UMTS AKA. In our comparison, there are five m values, namely, 2, 5, 10, 20, and 50. The x-axis is the bandwidth consumption measured in bits, and the y-axis is the number of registration within the same SGSN territory. The result is depicted as Fig. 9. From Fig. 9, we see that the bandwidth consumed by S-AKA is much less than UMTS AKA did.

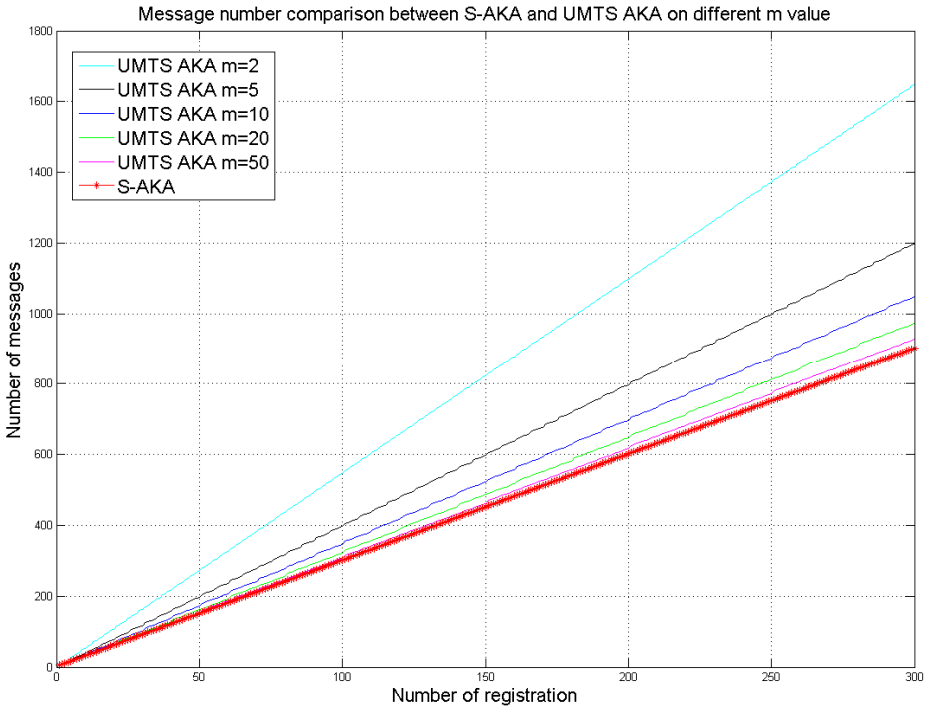


Fig. 10. Message number comparison between S-AKA and UMTS AKA

The message number comparison between S-AKA and UMTS AKA on different m values is shown in Fig. 10. The red thick line represents the number of messages of S-AKA. As we can see, our S-AKA uses the fewest messages.

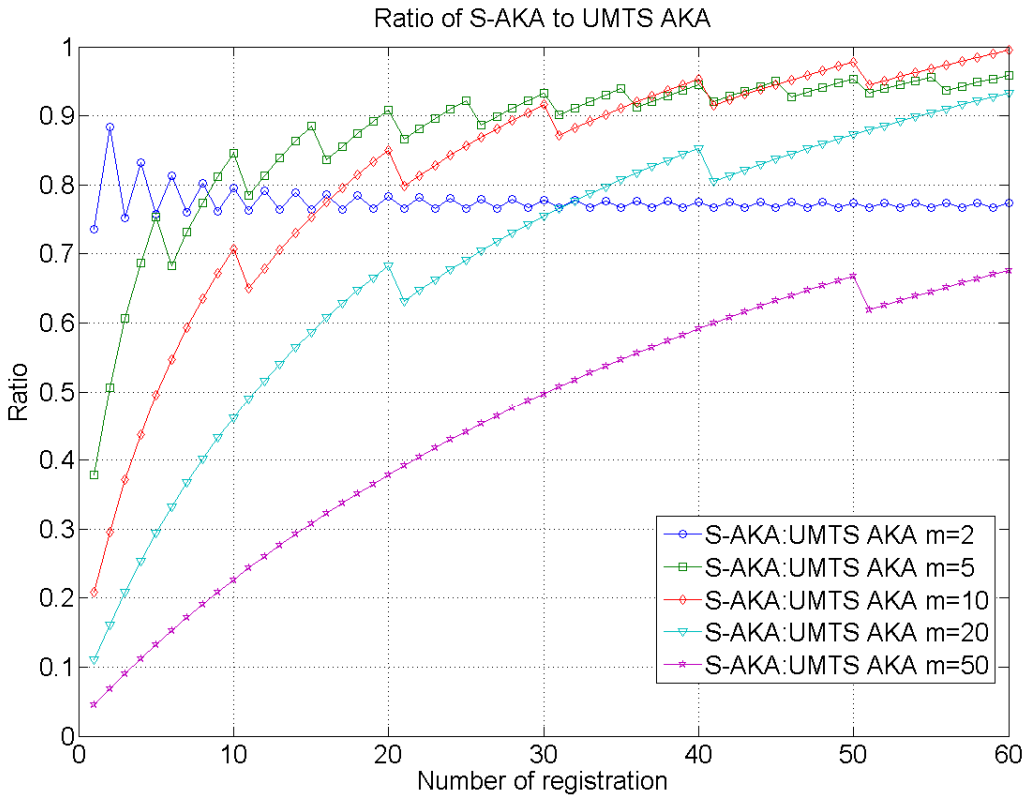


Fig. 11. The bandwidth ratio of S-AKA to UMTS AKA

Fig. 11 further illustrates the bandwidth consumption ratio of S-AKA to UMTS AKA. The x-axis of Fig. 11 is the bandwidth consumption ratio of S-AKA to UMTS AKA, and the y-axis is the number of registration within the same SGSN territory. As we can see, the lines of ratio are less than 1, which shows us the reduction degree on different m values. Table 3 shows some more detailed information about the comparison between S-AKA and UMTS AKA.

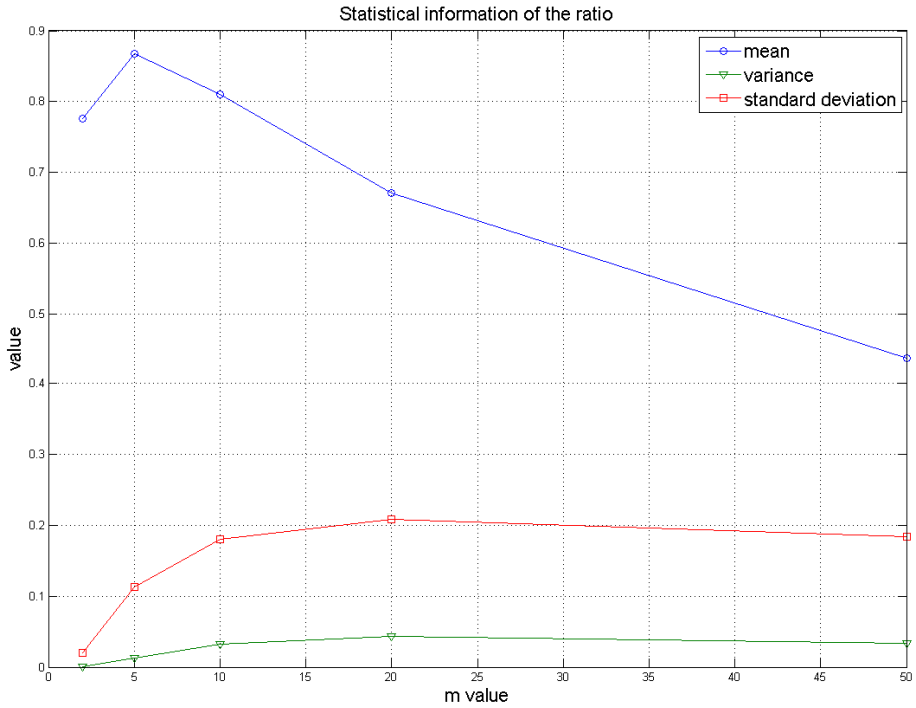


Fig. 12. The statistical information of the ratio

Fig. 12 shows the statistical information of the ratio computed in Fig. 11. In this figure, the x-axis represents the m value, and the y-axis is the value of the statistical information. The line with circle marker represents the mean of the ratio values on different m values, and the line with triangle marker represents the variance of the ratio values on different m values, and the line with cube marker represents the standard deviation of the ratio values on different m values. The statistical information is also listed in Table 3. And Table 4 lists the number of message of S-AKA and UMTS-AKA.

Table 3. The statistical information

m	**2**	**5**	**10**	**20**	**50**	**average**
the average ratio of S-AKA to UMTS AKA	**0.7755**	**0.8674**	**0.8102**	**0.6706**	**0.4366**	**0.7121**
variance	**0.0004**	**0.0129**	**0.0325**	**0.0434**	**0.0338**	**N/A**
standard deviation	**0.0207**	**0.1134**	**0.1804**	**0.2084**	**0.1838**	**N/A**

Table 4. The ratio of the number of messages

m (p=300)	2	5	10	20	average
the ratio of the number of messages (S-AKA : UMTS AKA)	0.549	0.7546	0.8622	0.9284	0.7430

As listed in Table 3, the average of the mean values on five different values is 0.7121, which means S-AKA reduced about 30% of the bandwidth consumption. And in Table 4, the average of the ratio is 0.7430, which means that S-AKA reduced about 25% of the number of messages.

5.3. Discussion

So far, we provided the security analysis and bandwidth analysis of the proposed S-AKA. Viewed in the security aspect, S-AKA not only fulfilled the security requirements of UMTS AKA but also provide further improvements, such as mutual authentication between SGSN and MS, defeating redirection attack and man-in-the-middle attack. On the other hand, when viewed in the efficiency aspect, S-AKA reduces the bandwidth consumption and the number of messages as well. S-AKA decrease the number of the messages transmitted between SGSN and HLR/AuC, which may be expensive when SGSN and HLR/AuC are located in different countries. With these security and efficiency features, it makes S-AKA a robust and economical authentication protocol for mobile networks.

6. Security proofs of S-AKA

In this chapter, we give security proofs of S-AKA to show that it is a secure authentication and key exchange protocol. We adopt Muxiang's security model [20] which is adopted from Shoup's formal security model [19] to prove our scheme. We first define some preliminaries, and then prove the security of S-AKA. For detailed description of the model, please refer to Muxiang's security model [20], and Shoup's security model [19].

6.1. 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 the function

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

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

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

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:

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

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, $Adv_G^{prf}(A)$ is negligible in k, then we say that G is a *pseudorandom* function family.

A *Message Authentication Code* is a family of functions $f1$ 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 tag or MAC of M. For the security of $f1$, we will use the notion of security against chosen message attacks. An adversary, called a forger in this context, 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 $Adv_F^{mac}(A)$, 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,

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

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, $Adv_F^{mac}(A)$ is negligible in k, we say that $f1$ is a secure message authentication code.

6.2. 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

$$\Pr(\overline{C_A}) \le \frac{n_i^2(2^{-|RAND|} + 2^{-|RAND_S|})}{2} \qquad (6.1)$$

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 $ID_{i'}$ 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 $PID_{i'j'}$ 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 use 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

$$\Pr(AS_{i'j'} = 1) \le \Pr(Succ(A_F, F)) = 1 \qquad (6.2)$$

And thus,

$$\Pr(AS_{i'j'} = 1) \le Adv_F^{mac}(t, q) \qquad (6.3)$$

, 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 $I_{p'j'}$ received the message (IMSI, FRESH, LAI, MAC_{MS}) before sending out AUTNs, where AUTNs = MACs||RANDs||RAND||AMF||FRESH, and MACs = $f1_{DK}(MAC_H||RAND||AMF)$. Since T_A is collision-free, RANDs 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 (6.3), we have

$$\Pr(IS_{ij} = 1) \le Adv_F^{mac}(t, q) \qquad (6.4)$$

,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 + n • RAND). Using the notation *Succ(A'F, F)* as described above, we have

$$\Pr(AS_{ij} = 1) \le \Pr(Succ(A_F^{'}, F) = 1) \qquad (6.5)$$

Therefore, by (6.4) and (6.5), the probability that the stimulus on a user instance I_{ij} was not output by a compatible network instance is upper-bounded by

$$\Pr(AS_{ij} = 1) + \Pr(IS_{ij=1}) \le 2 * Adv_F^{mac}(t,q) \qquad (6.6)$$

Case 3. Let $I_{i"j"}$ be a network instance which has received (XRES) and has accepted, where RANDs was sent out by $I_{i"j"}$ in the AUTNs. 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 (6.3), 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 AUTNs before it output the stimulus. Since T_A is collision-free, AUTNs can not be output by a network instance other than $I_{i"j"}$. This means that it is the adversary who concocted the MACs. By (6.6), 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,

$$Adv_{T_A, T_A^*}^{dist}(D) = Adv_G^{prf}(t,q)$$

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 which 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 ring master 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 transcript are computationally indistinguishable.

Case 1. Assume that a user instance I_{i1j1} has received the message (AUTNs) and has accepted, where AUTNs = MACs||RANDs||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 AUTNs 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} = $f1_{ki2}$(FRESH, MAC_{MS}). 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 = $f2_{Ki3}$(RANDs), RANDs 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 ring master set the session key $K_{i3',j3'}$ by K_{i3j3}.

The above analysis 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

$$Adv^{dist}_{T_A,T^*_A}(D) = Adv^{prf}_{G}(D')$$

Thus,

$$Adv^{dist}_{T_A,T_A^*}(D) = Adv^{prf}_{G}(t,q)$$

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 TA 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 TA. 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

$$Adv^{dist}_{T_A,T_A^*}(D) = \left|\Pr(D(T_A)=1) - \Pr(D(T_{A^*})=1)\right|$$

$$= |(\Pr(D(T_A) = 1 \mid M_A \cap CA) - \Pr(D(TA^*) = 1 \mid M_A \cap CA))\Pr(M_A \cap CA) + (\Pr(D(TA)$$

$$= \begin{vmatrix} \Pr(D(T_A)=1 \mid M_A \cap C_A) - \Pr(D(T_{A^*})=1 \mid M_A \cap C_A)) \Pr(M_A \cap C_A) + (\Pr(D(T_A)=1 \mid (\overline{M_A}) \cup (\overline{C_A})) \\ - \Pr(D(T_{A^*})=1 \mid (\overline{M_A}) \cup (\overline{C_A}))) \Pr((\overline{M_A}) \cup (\overline{C_A})) \end{vmatrix}$$

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

On the other hand,

$$\Pr(\overline{M_A}) = \Pr(\overline{M_A} \mid C_A)\Pr(C_A) + \Pr(\overline{M_A} \mid \overline{C_A})\Pr(\overline{C_A}) \leq \Pr(\overline{M_A} \mid C_A) + \Pr(\overline{C_A})$$

Therefore,

$$Adv_{T_A,T_A^*}^{dist}(D) \leq n_e Adv_G^{prf}(t,q) + \Pr(\overline{M_A} \mid C_A) + 2\Pr(\overline{C_A})$$

By (6.1), $\Pr(\overline{C_A})$ is negligible in k. By Lemma 1, $\Pr(\overline{M_A} \mid C_A)$ is also negligible. Hence, $Adv_{T_A,T_A^*}^{dist}(D)$ is negligible. S-AKA is a secure authentication and key agreement protocol.

7. Conclusion

In this paper, we first introduce the two security weaknesses of UMTS AKA, namely, redirection attack and man-in-the-middle attack, and the bandwidth bottleneck of UMTS AKA. Then we propose our scheme, S-AKA, which can defeat both redirection attack and man-in-the-middle attack and works more efficiently. We also provide security analysis and bandwidth analysis and compare UMTS AKA and S-AKA. In our analysis, our proposed S-AKA not only defeated those two attacks mentioned above, but also reduce up to 30% of bandwidth consumption and 25% of messages. And we also proved that S-AKA is a secure authentication and key exchange protocol.

8. Reference

[1] 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

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

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

[4] Chun-I, Pei-Hsiu Ho, and Hsin-Yu Chen. Nested One-Time Secret Mechanisms for Fast Mutual Authentication in Mobile Communications. (TWISC 成果發表, 2006)

[5] 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)

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

[7] Yi-Bing Lin, Ming-Feng Chang, Meng-Ta Hsu, and Lin-Yi Wu. One-Pass GPRS and IMS Authentication Procedure for UMTS. IEEE journal on selected areas in communications, June 2005

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

[9] 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.

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

[11] Zhi-Jia Tzeng, and Wen-Guey Tzeng. Authentication of Mobile Users in Third Generation Mobile Systems. Wireless Personal Communications, 2001

[12] Geir M. Koien, and Agder University College. An Introduction to Access Security in UMTS. IEEE Wireless Communications, 2004

[13] Bin Zhang, Jiang-Xing Wu. Authentication and Key Distribution Methods in Mobile Computing Environments. ICCNMC'03, 2003

[14] Alberto Peinado. Privacy and authentication protocol providing anonymous channels in GSM. Computer Communications, 2004

[15] Yan Zhang, and Masayuki Fujise. Security Management in the Next Generation Wireless Networks. International Journal of Network Security, Vol.3, 2006

[16] Chris J. Mitchell. The Security of the GSM air interface protocol. Technical Report, 2001

[17] Muxiang Zhang. Adaptive Protocol for Entity Authentication and Key Agreement in Mobile Networks. Internal Conference on Information Security Cryptology, 2003

[18] Mihir Bellare, and Phillip Rogaway. Provably Secure Session Key Distribution – The Three Party Case. Proceedings of the twenty-seventh annual ACM symposium on Theory of computing, 1995

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

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

[21] 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.

[22] European Telecommunications Standards Institute (ETSI). GSM 02.09: Security Aspects, 1993