# UMTS에서 인증 및 키 동의 프로토콜에 대한 분석

최용강*, 김대영*, 김상진**, 오희국*

*한양대학교, 컴퓨터공학과, **한국기술교육대학교 인터넷미디어공학부

# An Analysis of Authentication and Key Agreement Protocols in UMTS

Yonggang Cui*, Daeyoung Kim*, Sangjin Kim**, Heekuck Oh*

*Department of Computer Science and Engineering, Hanyang University
**School of Internet Media Engineering, Korea University of Technology and Education

## Abstract

In UMTS (Universal Mobile Telecommunication System), a protocol called UMTS AKA is used to authenticate MSs (Mobile Stations). When an MS is in a foreign network, the serving network contacts the AuC (Authentication Center) located at the home network of the MS to authenticate it. To reduce this cost, AuC sends *n AVs* (Authentication Vectors) to the serving network. Although the use of *AVs* allows the serving network to authenticate an MS without contacting the AuC each time, there are also shortcomings such as synchronization problem. Subsequently, a set of protocols adopting the same or similar method have been proposed. In this paper, we analyze and compare authentication protocols for UMTS with respect to the use of *AVs* and its alternatives. We conclude that using Kerberos-like ticket key overcomes some of the drawbacks of using *AVs*, whereas *AVs* provide much better security.

## I. Introduction

During the last few years we have observed steadily rapid development of wireless and mobile communication networks, especially in the UMTS (Universal Mobile Telecommunication System). The UMTS is one of the third generation mobile communication standards which is currently being launched throughout the world. And it inherits the framework of GSM (the Global System for Mobile) communications, its forerunner, which has many disadvantages [1-3]. Hence, the UMTS adopts an authentication and key agreement protocol, termed UMTS AKA. In the architecture, the UTRAN (tUMTS Terrestrial Radio Access Network) is connected to the MS via the radio interface, and communicates with the HLR (Home Location Register) and the AuC to receive subscriber data and authentication information of an MS, which performs mobility and data session management for GPRS mobiles and ciphering, compression of the data transmitted, and the routing of IP packets. The HLR/AuC may have precomputed the required *AVs* and retrieve the *AVs* from the HLR database or compute them on demand.

The purpose for generating *AVs* is to reduce the number of accessing AuC, since the cost for accessing AuC is expensive. Nevertheless, the *AVs* may occupy too much network bandwidth for transmission from the AuC to the SGSN (Serving GPRS Support Node). and subsequently

* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연센터(ITRC) 지원사업의 결과로 수행되였음

* 이 연구에 참여한 연구자는 '2단계 BK21 사업'의 지원을 받 았음.

presented protocol, such as AP-AKA [6, 8], Harn-Hsin [7], also employs the same or similar method, respectively. In this paper, we will analyze and compare the UMTS AKA, AP-AKA, Harn-Hsin and UMTS X-AKA protocols, and conclude that UMTS X-AKA is not superior to other protocols on security, and this protocol is similar to Kerberos protocol.

The remainder of this paper is organized as follows. Section II, section III and section IV briefly describes and analyzes the UMTS AKA, AP-AKA and Harn-Hsin protocols, respectively. In section V, we specify and discuss the UMTS X-AKA protocol and Kerberos protocol, And then compare UMTS X-AKA with Kerberos, and give a full comparison about each protocol from performance and security. In Section VI, we conclude the paper.

## II. UMTS AKA Protocol

### 1. Overview

The UMTS AKA protocol comprises two procedures which are the distribution of the *AVs* from HLR/HN to VLR/SN, and the authentication and key agreement between MS and VLR/SN. In the UMTS AKA protocol [8, 9], there are three entities, MS, VLR/SN, HLR/HN. MS and HLR/HN share a secret key and certain cryptographic algorithms. And both sides maintain a $SQN_{MS}$ counter and a $SQN_{HN}$ counter. In the following, we will give a description of the UMTS AKA protocol.

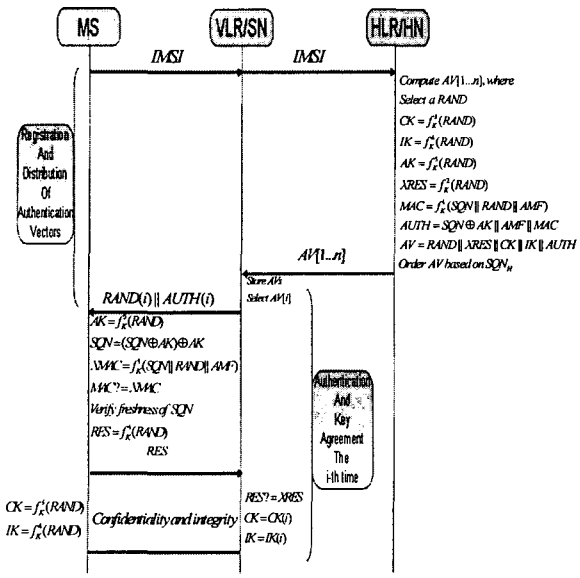


Fig. 1. UMTS AKA Protocol

As Fig. 1. shows, MS sends a authentication data request to HLR/HN via VLR/SN. On receipt of the authentication data request, the HLR/HN first of all generates a *RAND* (Random Number), and then computes *CK*, *IK*, *AK*, an expected response *XRES* and a *MAC* (Message Authentication Code). Next, the HLR/HN assembles the *AUTH* (Authentication Token) and aggregates the *RAND*, *XRES*, *CK*, *IK*, *AUTH* as *AVs* whose order is based on *SQN* [5], and then sends *AVs* to VLR/SN.

Subsequently, the VLR/SN selects *ith AV* from its database to initiate the procedure. The VLR/SN sends *RAND(i)* and *AUTH(i)* to MS. On receipt of them, MS will compute *AK* with *RAND* and retrieve the *SQN* thereby *AUTH*. Next MS will compare *XMAC* by computing with *MAC* received from the VLR/SN. If the result of comparison is not consistent, MS will reject authentication request. Otherwise the MS will further confirm whether the *SQN* is in the correct range. If MS finds *SQN* is not in the correct range, it will send a synchronization failure message back to VLR/SN [8]. Otherwise, MS computes $RES = f_K^3(RAND)$ and sends it back to VLR/SN.

### 2. Analysis

From the protocol, we can clearly see that MS can not authenticate VLR/SN which results in false base station attack. That is to say, an attacker can impersonate as a legitimate VLR/SN or HLR/HN to intercept connection request, then impersonate as MS to send connection request to a genuine VLR/SN or HLR/HN. As a matter of fact, the attacker relays information in the middle of MS and VLR/SN. In this way, MS can connect to an other network.

We can also see that UMTS AKA protocol possesses a salient feature. It employs *AVs*, namely, HLR/HN generates *n AVs*, and then distributes them to VLR/SN for subsequent authentication and key agreement procedure. Since the *AVs* will occupy too much bandwidth for transmission from the AuC to the SGSN, and the use of *AVs* will cover much storage space of the SGSN, so a storage space overhead occurs. In the protocol, using the *SQN* brings about synchronization problem.

# III. AP-AKA Protocol

## 1. Overview

Succedently, Zhang and Fang presented adaptive protocol for mobile authentication and key agreement, known as AP-AKA [6, 8]. As Fig. 2. shows that we will give a depiction of the AP-AKA protocol. AP-AKA execution in foreign network and in home network has a slight distinction. Typically, AP-AKA protocol consists of six flows in foreign network, while comprises three flows in home network.

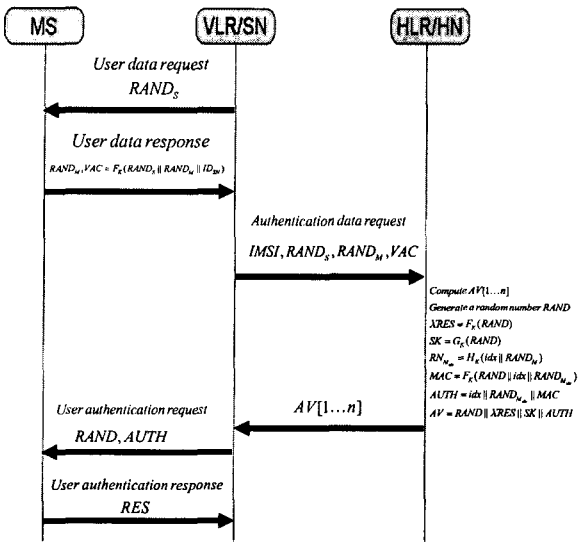


Fig. 2. AP-AKA Protocol

The AP-AKA protocol inherits the framework of the UMTS AKA protocol. Being analogous with UMTS AKA protocol, MS and HLR/HN share a secret key and certain cryptographic algorithms in the AP-AKA protocol which also comprises the distribution of the *AVs* and the authentication and key agreement.

At first, VLR/SN sends user data request to MS which is made up of a $RAND_S$. On receipt of the user data request, MS generates a $RAND_M$ and calculates *VAC*, then sends $RAND_M$ and *VAC* to VLR/SN. Next, VLR/SN sends an authentication data request to HLR/HN which is made up of *IMSI*, $RAND_S$, $RAND_M$ and *VAC*. After receiving the authentication data request, HLR/HN will verify the *VAC*, If the verification succeeds, HLR/HN will generate a *RAND* and distribute and index number *idx*, from *1* to *n*, and then compute an *XRES* (Expected Response) *SK*, $RAND_{M_{idx}}$ and *MAC*. Next, HLR/HN assembles the *AUTH* (Authentication Token) and aggregates the *RAND*, *XRES*, *SK*, *AUTH* as *AVs*, then sends *AVs* to VLR/SN.

After receiving the *n AVs* from HLR/HN, VLR/SN first of all selects *ith AV* from its database to mount the authentication and key agreement between MS and VLR/SN, then sends *RAND[i]* and *AUTH[i]* to MS. Upon receipt of user authentication request, MS will compute *MAC* and compare the one received from VLR/SN. If the verification fails, MS will reject the user authentication request, otherwise MS will calculate the *RES* and send it back to VLR/SN. By checking the *RES*, VLR/SN will authenticate MS.

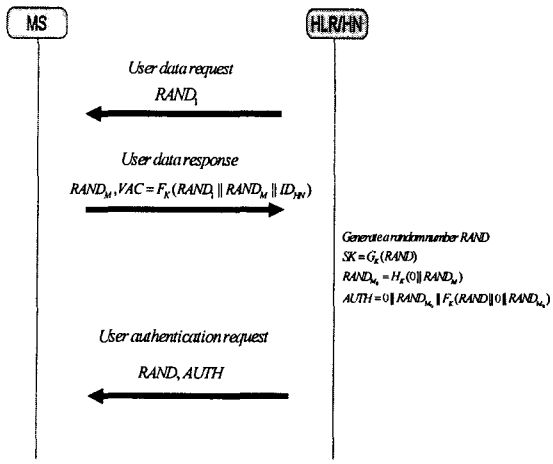


Fig. 3. AP-AKA Protocol in HN

When the protocol runs in the home work, as Fig. 3. shows that MS and HLR/HN implement the authentication and key agreement only by three flows [8].

## 2. Analysis

Compared to UMTS AKA protocol, AP-AKA protocol only appends two steps at the first, user data request and user data response between MS and VLR/SN, to verify that which VLR/SN requests the *AVs*. In the AP-AKA protocol, we can find that it eliminates the synchronization problem without using the *SQN*, and obtain the same effect as UMTS AKA protocol by using *idx* (Index Number).

Assume that an adversary wish to mount false base station attack in this protocol. At first, the

adversary will impersonate as a legitimate VLR/SN to get connection request, and then relay the connection request to a genuine SN. On receipt of it, VLR/SN generates a $RAND_S$ and sends it to the adversary. The attacker relays the $RAND_S$ to the MS. Subsequently, the MS generates a $RAND_M$ and calculates $VAC = F_K(RAND_S \| RAND_M \| ID_{HN})$, if the user is in the home network range. If the user is in the $SN_1$ range, MS will calculate $VAC = F_K(RAND_S \| RAND_M \| ID_{SN_1})$, then send $RAND_M$ and $VAC$ back to the adversary. The adversary relays them to SN, next SN sends $IMSI$, $RAND_S$, $RAND_M$ and $VAC$ to the HLR/HN. On receipt of these messages, HLR/HN computes $VAC = F_K(RAND_S \| RAND_M \| ID_{SN})$ and compares it with $VAC$ received from SN. We can easily find that $VAC$ computed by HLR/HN is not equal to $VAC$ calculated by MS. So the protocol prevents against the false base station attack. However, it employs the same $AVs$ method as UMTS AKA protocol, so bandwidth consumption and a storage space overhead will occur.

# IV. Ham-Hsin Protocol

## 1. Overview

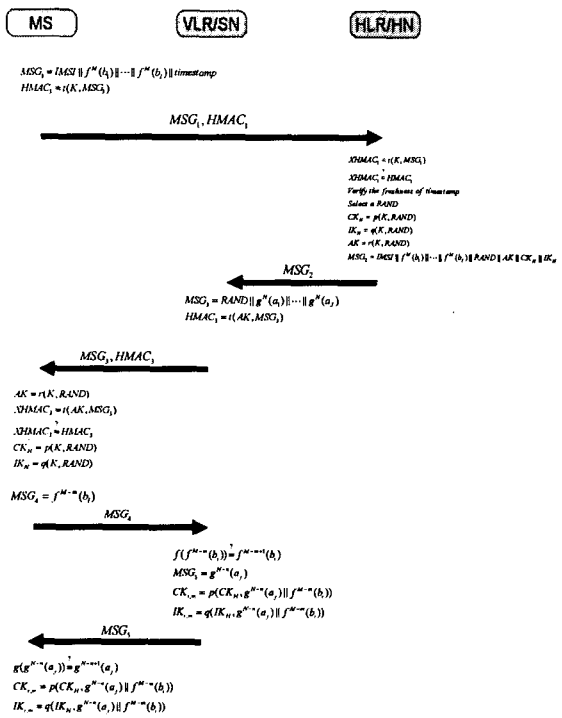


Fig. 4. Harn-Hsin Protocol

In the protocol, Harn-Hsin employs *HMAC* (Keyed-Hash Message Authentication Code) and hash chaining techniques. The MS can verify its own identity to the VLR/SN at most $I \times M$ times, while the VLR/SN can verify its own identity to the MS at most $J \times N$ times. And the VLR/SN has the local control in authenticating the MS, for example, VLR/SN can decrease the number of authentication times by reducing the number of random seeds generated by itself. The HLR/HN also has the total control by the use of the HLR-generated *CK* and *IK*. The *HMAC* and hash chaining method are simialr to the *AVs* method [7].

## 2. Analysis

In the Harn-Hsin protocol, MS generates $I$ random seeds to authenticate itself to VLR/SN. In my opinion, it is not meaningful for security and performance of the protocol, conversely, Using $I$ random seeds brings about bandwidth consumption problem and a storage space overhead and confusion how to select $f^{M-m}(b_i)$ in the authentication and key agreement procedure. Session keys in the mutual authentication procedure between MS and VLR/SN are not becoming strong, based on three input values from MS, VLR/SN, HLR/HN, respectively.

Compared to UMTS AKA protocol, AP-AKA protocol, we can easily find that Harn-Hsin has a stronger mutual authentication and non-repudiation by adopting the hash chaining technique. However, the protocol employs similar method as *AVs* in the UMTS AKA protocol, creating massive authentication information results in bandwidth consumption problem and a storage space overhead will occur.

# V. UMTS X-AKA Protocol

## 1. Overview

Finally, Huang and Li present UMTS X-AKA Protocol [9]. As Fig. 5. shows that UMTS X-AKA protocol adopted *TK* (Temporary Key) mechanism and solved the problems of bandwidth consumption, storage space and so on. First of all, MS will send $MAC_M, t, IMSI$ to HLR/HN via VLR/SN, here $t$ is timestamp. Then HLR/HN checks the $MAC_M$ and verifies the MS's identity. Next, HLR/HN computes the *TK*, $MAC_H$ and sends *TK*, $AUTH_H$ to VLR/SN. In the authentication and key agreement procedure the VLR/SN generates $RAND_S$ and computes $MAC_S$, then sends $AUTH_S$ to

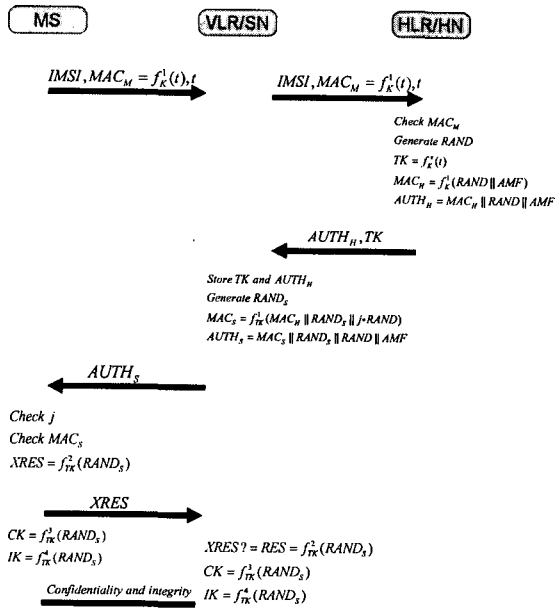


Fig. 5. UMTS X-AKA Protocol

MS. MS will check $MAC_S$ and send *RES* back to VLR/SN. On receipt of *RES*, VLR/SN will authenticate the MS's identity. Then both parties will compute *CK*, *IK* for subsequent security communications. To illustrate explicitly that UMTS X-AKA protocol is similar to Kerberos protocol, we will outline the Kerberos protocol [4] in the following.

Kerberos uses the Needham-Schroeder protocol as its basis. It makes use of a trusted third party, termed a KDC (Key Distribution Center), which consists of two logically separate parts: An AS (Authentication Server) and a TGS (Ticket Granting Server), Kerberos works on the basis of "tickets" which serve to prove the identity of users.

Kerberos maintains a database of secret keys; each entity on the network - whether a client or a server - shares a secret key known only to itself and to Kerberos. Knowledge of this key serves to prove an entity's identity. For communication between two identities, Kerberos generates a session key which they can use to secure their interactions.

what follows is a simplified description of the protocol. The following shortcuts will be used: AS=Authentication Server, TGS=Ticket Granting Server, SS=Service Server. In one sentence: the client authenticates itself to AS, then demonstrates to the TGS that it's authorized to receive a ticket for a service, then demonstrates to the SS that it has been approved to receive the service.

From the aforementioned content we can see that the UMTS X-AKA protocol is analogous with Kerberos protocol. It employs message authentication codes, MS sends $MAC_M$ to HLR/HN to authenticate itself to the HLR/HN, which is equal to the action that A (Alice) demonstrates itself to obtain the service that she wants to communicate with B (Bob) from the server. And then HLR/HN generates *TK* as a secret key between MS and VLR/SN which is set a lifespan in the communication session. As such this is consistent with server distributing a session $K_{AB}$ to A and B for subsequently secure communication. The unique difference is that both parties utilize the *TK* to compute a new *CK* and *IK*. Nevertheless, this is only a change in form but not in content.

## 2. Analysis

From aforementioned description of UMTS AKA, AP-AKA, Harn&Hsin, UMTS X-AKA protocols, we can easily obtain the following results as Table 1. shows.

Table 1. Comparision of Four Protocols in UMTS

| | UMTS AKA | AP-AKA | Harn-Hsin | UMTS X-AKA |
|---|---|---|---|---|
| UM | *AVs* | *AVs* | Hash Chain | *TK* |
| AT | *n* | *n* | $I \times M$, $J \times N$ | Session |
| MS | No | No | Yes | Yes |
| RBC | No | No | No | Yes |
| RSSO | No | No | No | Yes |
| NS | Yes | No | No | No |

AT: Authentication Times between MS and VLR/SN RSSO: Reduction of storage space overhead for VLR/SN's database
MA: Mutual authentication between MS and VLR/SN NS: Need synchronization between MS and HLR/HN

Note that MS can not identify VLR/SN's identity, which results in false base station attack in the UMTS AKA protocol. In the AP-AKA protocol, although MS can know who (which VLR/SN) requests *AVs* from the HLR/HN, MS still can not explicitly identify VLR/SN's identity. So MS and VLR/SN can not accomplish mutual authentication in the UMTS AKA and AP-AKA protocols in my opinion. In addition, Harn-Hsin protocol creates

massive authentication information which brings about bandwidth consumption problem and a storage space overhead. Bandwidth consumption problem should be consistent with a storage space overhead in this protocol, in other words, both problems should come about simultaneously. UMTS X-AKA protocol accomplishes mutual authentication between MS and VLR/SN, and eliminates bandwidth consumption problem and a storage space overhead by adopting *TK* method. However, this protocol has a lower security than UMTS AKA protocol. Since in case that the *TK* generated by HLR/HN is compromised, an adversary can directly compute session keys *CK* and *IK* used in authentication and key agreement. Conversely, UMTS AKA protocol can not come about such a problem. Since *CK* and *IK* are calculated by using random numbers generated by HLR/HN. In a word, these four protocols can not be argued as the perfect solution.

## VI. Conclusion

From the analysis and comparison, we clearly see that UMTS AKA, AP-AKA, Harn-Hsin, UMTS X-AKA protocols all adopt the same or similar method. UMTS AKA and AP-AKA use *AVs*, Harn-Hsin uses hash-chain, and UMTS X-AKA uses Kerberos-like ticket key. These three methods have advantages and disadvantages meaning that one of them cannot be argued as the outright solution. The *AVs* are independent to each other which mean that disclosure of a session key in an *AV* does not affect other *AVs*. However, this method consumes far more network bandwidth between home and the serving network than others. Although in Harn-Hsin, the AuC issues several hash chains, similar to *AVs*, a single long hash chain can be used. In this case, the network consumption required would be similar to methods using a ticket key. However, both ticket key and hash chain rely heavily on the ticket key or the root of the hash chain, respectively. Therefore, those using *AVs* offer better security.

### References

[1] L. Harn, H. Y. Lin, "Modifications to Enhance the Security of GSM," Proc. 5th Nat. Conf. Information Security, pp. 416-420 Taipei, Taiwan, R.O.C., 1995.

[2] H. Lin and L. Harn, "Authentication protocols for personal communication system," Proc. ACM Special Interest Group on Data Communications, pp. 256-261, 1995.

[3] C. H. Lee, M. S. Hwang, and W. P. Yang, "Enhanced privacy and authentication for the global system for mobile communications," Wireless Networks, pp. 231-243, 1999.

[4] B. Schneier, "Applied Cryptography," Second Edition, John Wiley& Sons, 1966.

[5] 3GPP TS 21.133. 3GPP Security; Security Architecture.

[6] M. Zhang, "Provably-Secure Enhancement on 3GPP Authentication and Key Agreement Protocol," Cryptology ePrint Archive, Report 2003/092, 2003.

[7] L. Harn, W.J. Hsin, "On the Security of Wireless Network Access with Enhancements," Proc. of the 2003 ACM workshop on Wireless security in 2003.

[8] M. Zhang, Y. Fang, "Security Analysis and Enhancement of 3GPP Authentication and Key Agreement Protocol," Wireless Communications IEEE Transactions in 2005.

[9] C. Huang, J. Li, "Authentication and Key Agreement Protocol for UMTS with Low Bandwidth," Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference in 2005.