An Effective AKA Protocol for UMTS*

Hsia-Hung Ou[†] Iuon-Chang Lin[§] Min-Shiang Hwang[‡§]

Department of Computer Science and Information Engineering [‡]
Asia University
500, Lioufeng Rd., Wufeng, Taichung 41354, Taiwan, R.O.C.
Email: mshwang@asia.edu.tw

Department of Computer Science and Information Engineering [†]
Hwa-Hsia Institute of Technology
111 Gong Jhuan Rd., Chung Ho, Taipei, Taiwan, R.O.C

Department of Management Information Systems [§]
National Chung Hsing University
250, Kuo Kuong Road, Taichung, Taiwan 402, R.O.C.

September 8, 2012

An Effective AKA Protocol for UMTS

Abstract

UMTS is the most popular third-generation mobile phone system at present, and the UMTS AKA is his authentication and key agreement protocol. There are some weaknesses in the existed UMTS AKA protocol, such as synchronization question, storage overhead, insecurity with leakiness authentication vector, and weakness in mutual authentications. In this paper, we use the concept of proxy key to design a more secure and more efficient UMTS AKA protocol. Our protocol not only resists above-mentioned drawback of the original UMTS AKA protocol on the security, but also can improve the performance of UMTS AKA protocol, including the computational overhead, storage overhead, and communication overhead.

Key Words: UMTS, AKA, Mutual Authentication

1 Introduction

Communication technology is continuously developing and improving. AMPS (Advance Mobile Phone System), the first analog mobile phone system, was developed by AT&T in 1983. It is generally referred to as the first generation (1G) of mobile telecommunications. The main weakness of the AMPS is that it is easily susceptible to eavesdropping. In 1990, the ETSI (European Telecommunication Standards Institute) proposed a digital mobile phone system which is known as GSM [11, 19] (Global System for Mobile Communication), the second generation (2G) of mobile telecommunication. The GSM is the most widespread mobile communication system currently in use. Since the bandwidth of GSM cannot cope with many new applications of mobile communications, several third generation (3G) mobile telecommunication systems have been proposed [12, 21, 30]. The UMTS (Universal Mobile Telecommunication System) is the most universally utilized among those proposed by 3GPP [25].

*This work was supported in part by Taiwan Information Security Center (TWISC), National Science Council under the Grants NSC 96-2219-E-001-001, and NSC 96-2219-E-009-013.

§Corresponding author.

The UMTS AKA (Authentication and Key Agreement) [5, 8, 10, 18, 22, 28] protocol was established for 3GPP to solve authentication and key agreement problems. The key point of UMTS AKA protocol was uses a key which is shared between the USIM (Universal Subscriber Identity Module) on the cell phone and the AuC (Authentication Centre) in the user's HE (Home Environment) to achieve mutual authentication by both the user and the network. Moreover, the USIM and the HE keep track of the counters' SQN (Sequence Number) respectively to support network authentication and prevent replay attack [24, 26].

Many papers [9, 13, 29] have stated that only asymmetric cryptography can solve the security threats of the third generation mobile phone system. They used the techniques of asymmetric cryptography, such as the RSA [27], the PKI [7, 14], and the certificate [20] to solve authentication and key establishment issues. Asymmetric cryptography compared to symmetric cryptography demands more computational costs. Therefore, if it is employed, mobile equipment must be expected to have the calculation ability to meet these demands. Thus, although asymmetric cryptography has many advantages in authentication and key establishment, we do not discuss these in this paper. We put object at currently UMTS standard related technique. Because what we want is a system can be implementing immediately rather than a system that is waiting for future progress.

Harn and Hsin proposed an enhanced registration and AKA procedure in 2003 [15] called ER-AKA protocol. The main characteristic of ER-AKA is to apply the hash-chain [23] for authentication. But the MS and the SN must store the lot of hash-chain which generated by itself and the other side. Current mobile equipment storage space is insufficient. So, it wills additional loader of mobile equipments and cause to practice is difficulty.

In 2005, Zhang and Fang proposed an adaptive protocol for mobile authentication and key agreement, called AP-AKA [31] protocol. Their protocols primarily focused on working out the SN camouflage problem. Their idea was to enclose both identity and randomness in the message authentication code. But their methods only help the HE solve problem of the validation authentication request. Also, in other aspects, it cannot increase safety. The related reasons will discuss behind the section.

Hung and Li have proposed an UMTS X-AKA protocol [17] in 2005 afterward. X-AKA shifts the focal point to the bandwidth consumption and the space overhead. They proposed a temporary key concept improves the insufficiency of original UMTS AKA protocol. However, X-AKA protocol also has some may improve. There are, some key generation function and message authentication code function cannot found on the 3GPP standards and some MAC usages are not necessary.

In 2009, Hu et al. [16] exchanged the original role of the work in the UMTS AKA protocol and proposed a new UMTS AKA protocol aimed at less delay. The protocol does not use the authentication vector, MS calculates an authentication token, HE calculation of the corresponding Res, as is the SN responsible for verification. However, despite of such a way that reduces the fixed network traffic but increased traffic for mobile devices.

Through the literature, some drawbacks can be found in the UMTS AKA protocol and listed as follows:

1. Synchronization difficulty with SQN:
   SQN is a sequence number which is kept and maintained in both MS (Mobile Station) and its HE. The purpose of originality is to prevent replay attack, but maintenance consists with the SQN is difficult. Every faulty transmission or authentication will lead it to be asynchronous.

2. Strong storage requests on the SN (Serving Network):
   In the UMTS AKA protocol, when an MS enters a service scope of SN, SN gets n sets of AV (authentication vector) from user's HE and stores it for authenticate the MS on the later. When a lot of MS come together in an SN's service, the SN needs many storage spaces.

3. The authentication vector (AV) leakiness will cause insecurity:
   AV included n sets of related parameter that can represent HE to mutual authenticate with MS. It is generated by the MS's HE and transmitted to the current service network (SN) while a MS enter an SN's service. When the MS roams around in different service networks, the unused authentication vectors will transfer along with the MS's position to the current service network. Thus, the risk of AV disclosure is raised.

4. Weakness for mutual authentication:
   Observing the UMTS AKA protocol, one will find that the SN did not yield a unique status with MS. It uses the AV that the HE provides to win the MS's confidence. So, MS can only ensure that the AV is correct, but cannot promise that the SN is legal.

In order to solve the weakness of the UMTS AKA protocol, we propose an improved protocol to deal with authentication and key agreement problems in the UMTS protocol. We will first introduce the UMTS AKA protocol in Section 2. Then our improved protocol is described in Section 3. The computational overhead, storage overhead, and communication overhead of ER-AKA, AP-AKA, and our scheme are discussed and compared thereafter. Finally, the conclusions are presented in the last session of this paper.

2 Overview of the UMTS AKA Protocol

The UMTS AKA protocol (Shown as Figure 1) can be divided into two phases. One is the distribution of authentication vectors phase, another is authentication and key establishment phase. The distribution of authentication vectors phase is generates and delivers n sets of authentication vectors (AV) from the user's HE/HLR (Home Location Register) to VLR (Visitor Location Register)/SN to perform n times of user authentications. When the MS arrives in SN or VLR, it sends an IMSI (International Mobile Subscriber Identity) or TMSI (Temporary Mobile Subscriber Identity) to SN and the SN delivers an IMSI to the user's HE. HE then generates n sets of authentication vectors and sends back to the SN for authentication of the MS later. The produced are as follows where $RAND$ is a random number, k is a secret key shared between the MS's USIM and the MS's HE, $f1 \sim f5$ are the key generating function or authentication function [1, 2, 3].

1. Cipher key $CK = f3_K(RAND)$.
2. Integrity key $IK = f4_K(RAND)$.
3. Anonymity key $AK = f5_K(RAND)$.
4. Expected response $XRES = f2_K(RAND)$.
5. Message authentication code $MAC = f1_K(SQN||RAND||AMF)$, where SQN is a sequence number which maintains consistency between MS and HE, and AMF (Authentication and Key Management Files) is used to indicate the algorithm and key used to generate a particular authentication vector.
6. Authentication token $AUTH = SQN \bigoplus AK||AMF||MAC$.
7. Authentication vector $AV = RAND||XRES||CK||IK||AUTH$. Repeat the above mentioned action until n sets of AV are produced.

In the authentication and key establishment phase. The purpose of this phase is to authenticate the user and to establish a new pair of cipher and integrity keys between the MS and the SN. When the SN gets the authentication vectors from the HE or another SN, it uses the next unused authentication vector in the order array of AV to authentication MS. SN gets $RAND$ and $AUTH$ from AV then delivers

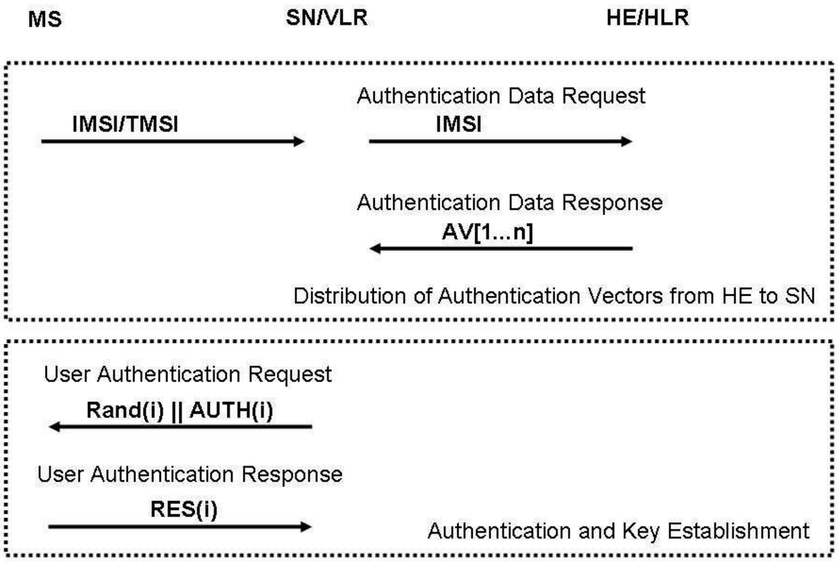

Figure 1: The UMTS AKA Protocol

it to MS. MS uses the $RAND$ and the secure key K which is shared between MS's USIM and MS's HE to compute the following:

1. Cipher key $CK' = f3_K(RAND)$.
2. Integrity key $IK' = f4_K(RAND)$.
3. Anonymity key $AK' = f5_K(RAND)$.
4. Response $RES = f2_K(RAND)$.
5. Verify the MS's sequence number SQN' while equal $AK' \bigoplus (SQN \bigoplus AK)$ within AUTH.
6. Calculate $MAC' = f1_K(SQN'||RAND||AMF)$.
7. Verify MAC' while equal MAC.

If the SQN and MAC are positive, the MS confirms that the SN is a legitimate service network. In the above mentioned steps, the MS will authenticate the SN then sends back a RES which it calculates in above Item 4, to the SN. The SN compares the RES with the $XRES$ within $AUTH$. If they are equal, it indicates that the MS is a legal user.

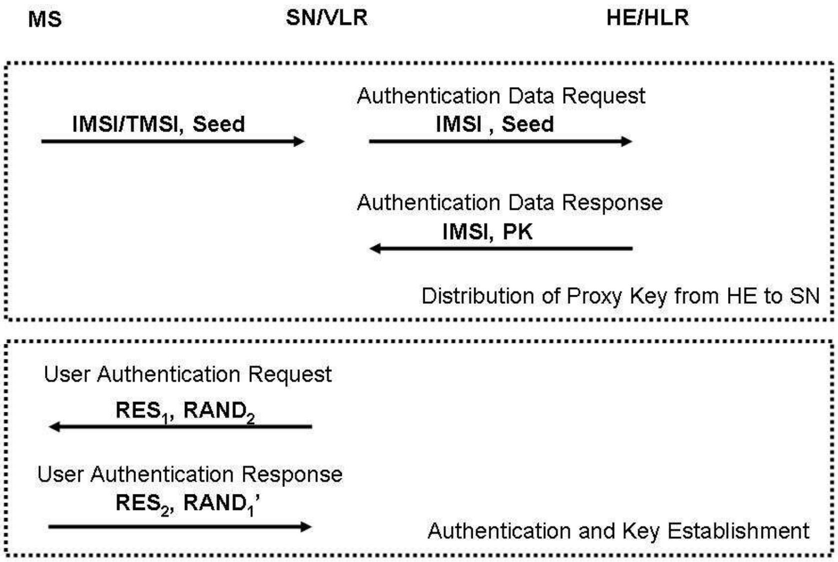

Figure 2: Our Enhanced UMTS AKA Protocol

3 The Proposed UMTS AKA Protocol

Our main idea makes use of the proxy key. For working out the program of space overhead on the SN, we used a proxy key to replace authentication vectors. The proxy key is produced by the HE and used to authenticate and agree keys with the MS. Thus, the SN does not need to hold the n sets of the authentication vectors. Our protocol is not the same as the UMTS AKA protocol, SN must prove that it is legal to the MS. Figure 2 illustrates our improved protocol.

As Figure 2 shows, two phases can be found. One is the distribution of proxy keys from the HE to the SN. Another is the authentication and key establishment. The details of the two phases are described as follows.

3.1 Distribution of Proxy Key from the HE to the SN

The purpose of this phase is to produce a proxy key and deliver the proxy key from the HE to the SN. The proxy key will be used for authentication between the SN and the MS in the future. The first time that the MS arrives at an SN service scope, this phase will be enforced. Details given as follows:

1. The MS randomly generates a 128-bit number as a *Seed*.
2. The MS calculates the $PK = f4_K(Seed)$. K is a secure key which is shared between the MS's USIM and the MS's HE.

3. MS deliver $(IMSI, Seed)$ to SN.

4. The SN pass forward the $(IMSI, Seed)$ to MS's HE.

5. The HE also calculates $PK = f4_K(Seed)$ and sends it back to the SN. The K is a secure key which is shared between the MS's USIM and the MS's HE.

6. The SN takes $Seed$ as the $RAND1$ and stores the $(IMSI, PK, RAND1)$ on its side.

3.2 Authentication and Key Establishment

If the MS moves within the scope of the SN, and the above mentioned phase has be enforced, then this phase will be enforced once. Details are listed as follows.

1. The SN calculates the $RES_1 = f2_{PK}(RAND_1)$, and randomly generates a 128-bit number as the $RAND_2$.

2. The SN delivers the $(RES_1, RAND_2)$ to the MS

3. The MS calculates the $TRES_1 = f2_{PK}(RAND_1)$ and compares the $TRES_1$ with the RES_1. If equality indicates that the SN is legal, otherwise this connection will break down.

4. The MS calculates the $RES_2 = f2_{PK}(RAND_2)$, and randomly generates a 128-bit number as the $RAND_1'$.

5. The MS delivers the $(RES_2, RAND_1')$ to the SN.

6. The SN calculates the $TRES_2 = f2_{PK}(RAND_2)$ and compares the $TRES_2$ with the RES_2, if equality indicates that the MS is legal, otherwise break down this connection.

7. The MS and SN calculates a cipher key $CK = f3_{PK}(RAND_2)$ and a integrity key $IK = f4_{PK}(RAND_2)$.

8. The MS and SN store the $RAND_1'$ and replace it with the $RAND_1$, and using CK and IK to encrypt their communication.

3.3 Discussion and Analysis

Our AKA protocol is designed on the basic principle of the UMTS AKA protocol. Taking realistic situations into account, there are no additional mechanisms in our protocols, such as extra encryption algorithms or key generation functions. Some points for the UMTS AKA protocol are as follows.

1. Uses a secure key shared between the MS's USIM and the MS's HE to achieve mutual authentication between the user and the network.

2. Uses the challenge/response protocol to achieve maximum compatibility with the current GSM security architecture.

3. Keeps track of a sequence number SQN respectively by use of the USIM and the HE to support network authentication.

4. Compares the counter of the SQN to confirm that the authentication data is fresh to prevent replay attacks.

5. Establishes different cipher key CK and integrity key IK each time, ensures that communication contexts are kept secret and integrity maintained.

Regarding the previously mentioned five points, we will illustrate in the following four points further improvements that our enhanced protocol can achieve:

1. In addition to the original secret key that is shared between the MS and the HE, we include a proxy key PK. HE authorizes the proxy key PK to the SN to authenticate the MS. The mutual authentication between MS with SN on the UMTS AKA protocol is based on the authentication vector AV which is to be generated by the HE and usage by the SN. The SN can use the $RAND$ and $AUTH$ within AV to demonstrate its legitimacy for the MS, and verify the response from the MS to confirm MS's legitimacy. Therefore the MS is mutually authenticated with the HE. The role of the SN is just an extension of the HE in this case. The MS can only trust the SN which owns the AV but cannot verify the SN. In the UMTS standard [4], the AV alone does not merely hold the SN. When the MS roams over the other SN, the surplus unused AVs will also follow its transfer. This will result in a very high risk of the AV leaking. Our method resists this program, we don't use the AV to carry out the authentication, but rather allow the HE to authorize the SN a proxy key which the MS also can infer it. Based on the proxy key PK, mutual authentication can proceed. Moreover, unlike the AKA protocol which requires more storage to store the n sets of the AV on the SN, we only need to store a proxy key PK on the SN. Therefore, our protocol does not transmit the PK to other service networks. When the MS leaves the current service network, establishing a fresh PK again is necessary. Please notes the PK was generated by $f4_K(Seed)$ and the $Seed$ is select by MS. So the PK isn't that the valid permanence. Since the MS roaming around to a new SN or communication over the limit times, the distribution of proxy key phase will be rerun and the PK will be re-drawn

up. If there is any malicious attacker sent the MS's IMSI and Seed SN in order to attempt and influence the connection between MS and SN, it will not be succeed. Because when MS finds the connection problems, it will re-send its IMSI and Seed to re-create a new PK. Since our protocol governing the phase of the distribution proxy key does not need a very large computational overhead and storage space, the capability question will not take place.

2. In the original UMTS AKA protocol, the SN authenticates the MS by the challenge/response method, and the MS authenticates the SN by the message authentication code (MAC) that generated by the HE then pass to SN. In our enhanced protocol, both the MS and the SN mutually authenticate by the challenge/response method. In other words, our enhanced protocol enables immediate authentication between the MS with the SN, rather than mediate authentication between the MS and the MS's HE on the original UMTS AKA protocol. The advance is it can achieve genuine mutual authentication, and the both sides have the fair treatment. Moreover, the stronger mutual authentication is achieved.

3. SQN must be securely shared among the MS and the HE in the original UMTS AKA protocol. The MS compares its SQN with the SQN inside the $AUTH$ to verify the legitimacy of the service network. This is an easy method but have a question how to maintain the correct SQN. Maintaining the SQN can cause asynchronous question because of the problem of network transmission. In addition, three SQN counters, the IMSI, SN and the HE are being maintained separately on the UMTS AKA protocol. Furthermore, there is operational difficulty with maintaining consist with SQN. Although they have sub-protocols that reset each other's SQN, every time this resetting occurs, the distribution authentication vector must be enforced again. This will increase the overhead of network. Our authentication method is based on the double challenge/response method. Synchronous programs are not occur in our enhanced protocol.

4. The UMTS AKA protocol checks the counter of the SQN to verify the replay attack. Because the fresh $AUTH$ within AV must have a bigger counter than the MS's SQN. But maintaining the consistency of the SQN is difficult as the explanation above demonstrates. In our proposed, to prevent replay attacks, each time the phase of authentication and key establishment is passed, we will in advance establish the $RAND$ value for the next time. This has same function with SQN, but more easily and only deliver between MS with SN, so

the question of consistency can to ignore.

In conclusion, our enhanced protocol has some advantages over the UMTS AKA protocol. These are stated as follows.

1. Strong mutual authentication: This has been already introduced above. The mutual challenge/response method established in our protocol are better than the one way authentication method in the UMTS AKA protocol.

2. Low storage space: In our protocol, the SN only holds a proxy key PK. This can perform a certification of infinite time while the proxy key PK is being established. The original UMTS AKA protocol of the SN must hold n sets of AV, and can only certify n time. Our PK can be used to authenticate unlimited times, but we suggest limiting its suitability because the long term key can easily attain the security threshold.

3. Low computational overhead: The HE does not require pre-computing authentication vectors in our protocol. In addition, it simplifies the network authentication method by replacing message authentication codes with challenge/response. Also, The MS reduces computational overhead.

4. Low network bandwidth: Since authentication vectors do not meet in our protocol, it is not necessary to deliver great authentication vectors over the networks whether HE deliver to SN or SN deliver to next SN.

4 Comparisons

In the above section, we already introduced the UMTS AKA protocol and our protocol. In this section, we will first simply explain before introducing this article of ER-AKA and AP-AKA. Then compare these protocols and our protocol of merits and shortcomings.

4.1 ER-AKA, AP-AKA and X-AKA Protocol

Figure 3 is the ER-AKA protocol. Their characteristic is uses the hash-chain and the MAC (Message Authentication Code). In the registration and distribution of the authentication information phase, the MS usage MAC proves his status to the HLR. The VLR also uses the MAC to prove his status to the MS. Two hash chain sets b_i and a_j were hidden in their MAC to ensure its validity. The VLR takes the $RAND_H, AK, CK_H$, and the IK_H that the HLR delivers to be the foundation

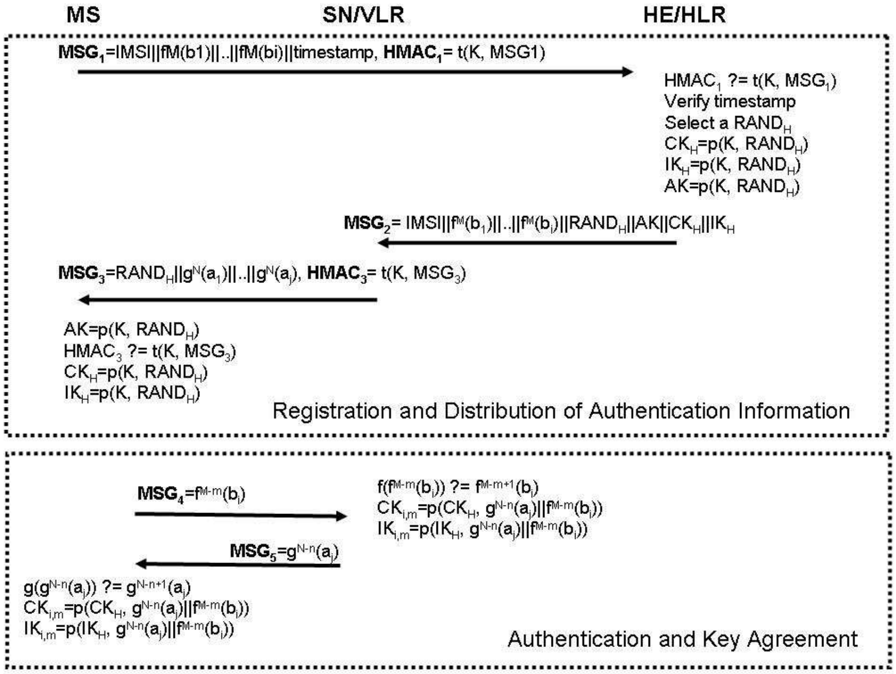

Figure 3: The ER-AKA Protocol

for authenticate with the MS. In the Authentication and key agreement phase, the MS and the VLR use the hash chain to authenticate each other. They explain that their protocol enhancements provide these main advantages: strong mutual authentication, strong key agreement, and non-repudiation of service. However, we do not agree with all their viewpoints. Firstly, non-repudiation is cannot be reached. Because of the VLR and the MS obtains all hash-chains on the initiation. Therefore, they can reciprocally provide proof for each other at every time. Secondly, key agreement has no special strength, much as the UMTS AKA protocol, their session key is different each time. In the other case, the MS must store a hash-chain which it generated by itself and the VLR. Since the mobile equipment has only a limited space, it will additionally space overhead on the mobile equipment.

The AP-AKA protocol is illustrated as Figure 4. They enumerated three problems in the UMTS AKA protocol. They are as follows: redirection attack, active attack in corrupted networks, and operational difficulty with sequence numbers. The first two problems are occurrences in the false BS (Base Station). Although this kind of situation does not easily occur in reality, the authors attempted to work out this problem. Their idea was to enclose both identity and randomness in the message authentication code. But their methods only help the HE work out the problem of the validation authentication request, MS and SN cannot get a guarantee. When the MS roams through other service networks, it has the unused authentication vectors.

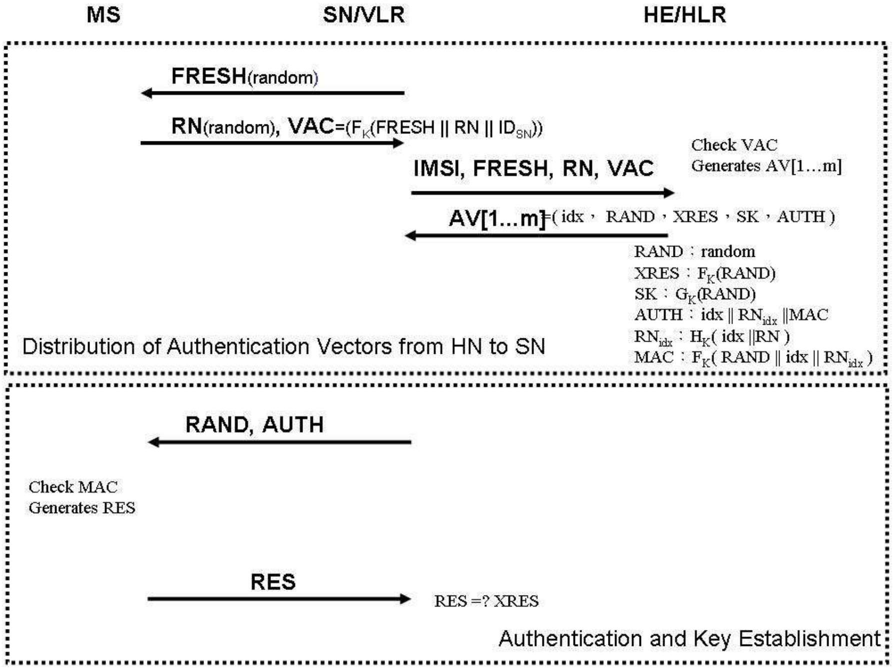

Figure 4: The AP-AKA Protocol

It will deliver to the new service network from last SN to next SN. The problems of false BS will still take place. Moreover, they use a random number and an index number to replace a sequence number (SQN) to solve the synchronous problem. In fact, the problem of synchronicity is not solved. Because their methods are similar to original SQN method, all of them have the program of sequence.

Unlike ER-AKA and the AP-AKA protocol, X-AKA protocol is focus on the performance. Hung and Li thought that UMTS AKA protocol has 3 questions to await the improvement. There are, 1) bandwidth consumption between SN and HN, 2) space overhead of SN, and 3) sequence number (SQN) synchronization problem. In order to solve the programs, they have substituted a temporary key mechanism for the sequence number determination. SN uses the temporary key which HE granted to replace the original authentication vector to mutual authentication with MS. Because has cancelled the authentication vector, therefore does not need to deliver and to store up it, and also to solve the shortcomings of the UMTS AKA protocol. Figure 5 is the X-AKA protocol. It looks like well but it is not perfect. First, the temporary key generation function f_x cannot be found among the 3GPP standard. And their message authentication code function f1 usage is different with the 3GPP standard. These cause some compatibility problems for the X-AKA protocol in real-life applications. Otherwise, the message authentication code MAC_u is necessary in the protocol. It is a redundant procedure because a false random number t cannot

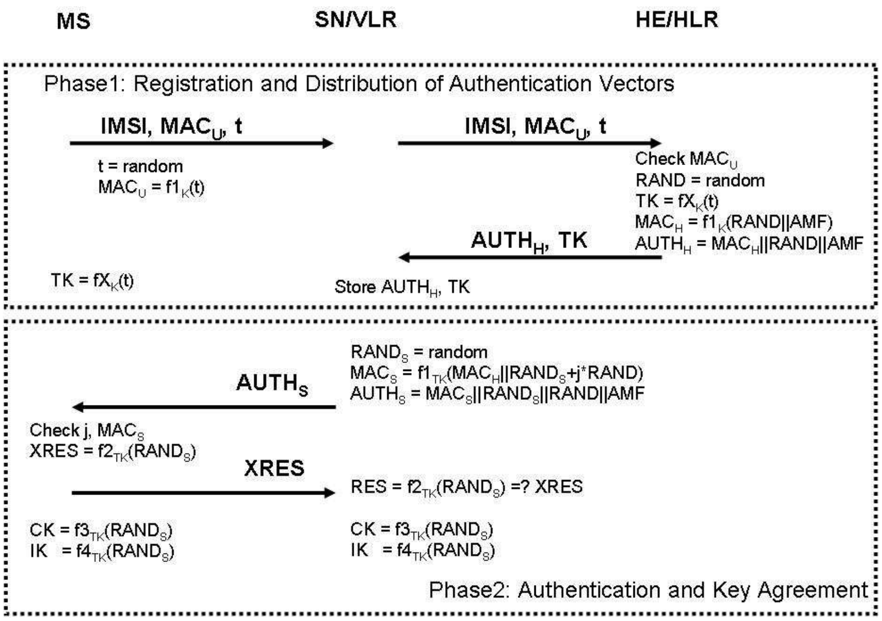

Figure 5: The X-AKA Protocol

pass the identification on the protocol. Moreover, our protocol is better than their performance. This will obtain the confirmation in afterward section.

4.2 Computational Overhead

In the mobile communication environment, mobile equipment usually doesn't have the height computing capability. Since the computational overhead will decide the practicability of the UMTS AKA protocol. There are three main members in the mobile communication environment; they're the MS, SN and the HE. Among them, only the MS is mobile equipment, the SN and the HN is the fixed facilities. Therefore, the computational overhead on the MS has to as much as possible reduced. And SN and HN can share one more.We collect and compare the computational overheads of these AKA protocols in Table 1. We cannot value their computational overhead because the different encryption function can't to compare. So we list all encryptions functions for once using on the table. In order to more clearly understand the differences between the protocols, different encryption functions are simplifies as a calculation unit for calculating their computational overload on the table.

In the UMTS AKA protocol, the computational overhead carries on the different members at a different stage. The HE shares the all computational overhead at the distribution of authentication vector phase. As well as the MS shares the all

Table 1: The Comparison of Computational Overhead

	UMTS-AKA	ER-AKA	AP-AKA	X-AKA	Our proposed
Computational Overhead on the Phase 1 (Distributed of authentication vector(Proxy key) Phase)					
MS	None =0	R×2+H×m+t×2+r+p+q =7+m	R+F =3	G+f1+fk =2	r+f4 =2
SN	None =0	R+H×n+t =2+n	R =1	None =0	None =0
HE	(R+X+f1+f2+f3+f4+f5)×n =7×n	R+t+p+q+r =5	(R+F×2+G+H)×n =5×n	G+f1×2+fk =4	f2 =1
Computational Overhead on the Phase 2 (Authentication and Key Establishment Phase)					
MS	X+f1+f2+f3+f4+f5 =6	H+p+q =3	F×2+F×2 =4	f1×2+f2+f3+f4 =5	f2×2+f3+f4 =4
SN	None =0	H+p+q =3	None =0	G+f1+f2+f3+f4 =5	f2×2+f3+f4 =4
HE	None =0	None =0	None =0	None =0	None =0

computational overhead at the authentication and key establishment phase. The SN is only delivering and comparing the message. For this reason, the HE must compute all authentication vectors for future authentication needs and the MS also compute these vectors for authentication. This method is more extreme with other methods. Whether the HE at the distribution of authentication vector phase or the MS at the authentication and key establishment phase, they share the computing overhead more than the other protocols. The AP-AKA protocol is similar to the UMTS AKA protocol. So, it also has the same condition. Different from UMTS AKA protocol, AP-AKA protocol increased some computational on the MS and the SN in the distribution of authentication vector Phase for preventing the false SN.

The ER-AKA protocol uses a different technique on the authentication. The advantage is their method that compared hash value in the authentication and key establishment phase can have the less computational overhead. But generating those hash chains is much cost at the distribution of authentication vector phase.

Our proposed is similar with X-AKA, but the efficiency is better. Moreover, our protocol is more efficient than these protocols. In the distribution of authentication vector phase, we have the best effect. Although we increased some computational overhead in the phase of authentication and key establishment but unlike the UMTS AKA and AP-AKA we equally shared it with the SN and the MS. So, one of them will not have the computational that can't bear. Furthermore, the UMTS AKA and AP-AKA have no real provide mutual authentication in the authentication and key establishment phase. This is a reason why their computational overhead is less than

Table 2: The Comparison of Storage

	UMTS-AKA	ER-AKA	AP-AKA	X-AKA	Our proposed
MS	SQN =48	b×(m+1)+a×(n+1)+CK+IK =256+Size(b×(m+1)+a×(n+1))	Rand×2+idx =304	t+TK+j =306	PK+Rand =256
SN	AV×n =608×n	((b×(m+1)+a×(n+1)+CK+IK+Rand) =384+Size(b×(m+1)+a×(n+1))	AV×n =608×n	$Auth_H$+TK+j =416	PK+Rand =256
HE	SQN =48	None =0	idx =48	None =0	None =0

our protocol in this phase. The protocol of our AKA and ER-AKA have already attained real mutual authentication. Since the computing characteristic of hash chains is easy and fast, our computing overhead is worse than the ER-AKA protocol in authentication and key establishment phase. But our protocol is better than their protocol with the computing overhead in the distribution of authentication vector phase.

4.3 Storage Overhead

Table 2 illustrates the comparison of storage overhead. Obviously, the ER-AKA protocol needs more space and our protocol was obviously won in this comparison. The UMTS AKA protocol and the AP-AKA protocol must store n sets of authentication vector on the SN for authentication with the MS later. Our protocol and ER-AKA protocol uses the idea of agents. SN authenticates the MS does not used the authentication vector that the HN gives, but a proxy key that the HN gives was been used. Because the ER-AKA protocol used the technique of hash chains, so it has to store more data although the effect of hash chains is better. Both X-AKA and our protocol can reduce a lot of storage overhead on MS and SN. This is a great help while the SN was work busyness and the storage ability is limited on MS. According to the Table 2 comparisons, our proposed has the lower storage space.

4.4 Communication Overhead

The other matter on compared with those protocols is the communication overhead. Our compared method is the bits number that total delivers. The bits number of key and relate data on the UMTS AKA protocol show as Figure 6.

Both the UMTS AKA protocol and our protocol follow the 3GPP standards so it can easily calculate the total bits numbers. AP-AKA protocol and the ER-AKA protocol uses the private key generation and encryption function which defines

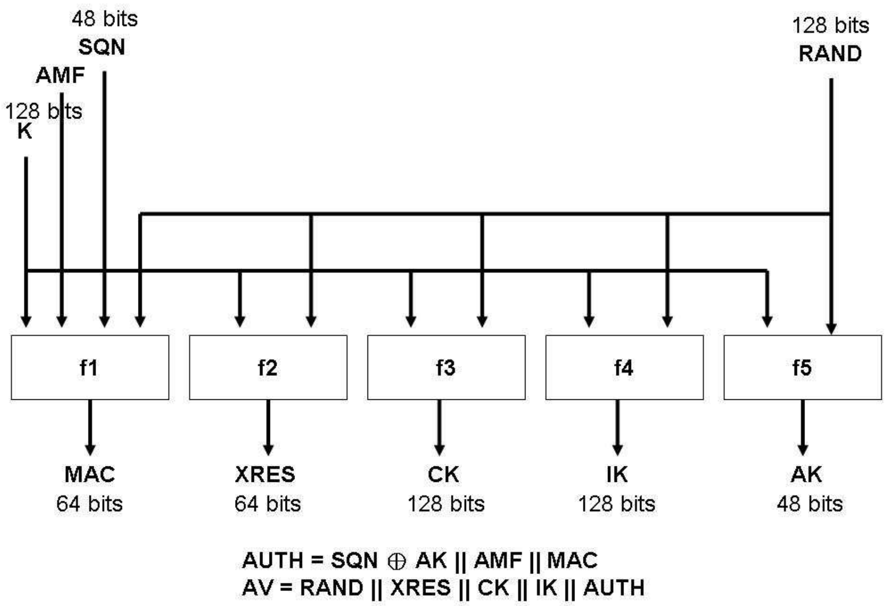

Figure 6: The Bits Number of 3GPP Key Generation Function

themselves and do not have explicit bits number. So we contrasted their private functions with the same capability function on the 3GPP standard to calculate their bits number. Some values that could not be found with contrast, we will directly indicate. Table 3 shows our conclusion.

Compared our protocol with other protocols on communication overhead, our proposed is smallest than other protocols at the phase of distribution of authentication vector. In the authentication and key establishment phase, our traffic is just about the same as other protocols. Obviously we came out on top again.

5 Conclusions

Most of the improved AKA protocols use the other cryptographic algorithms which designed by themselves. We do not think this is a good idea because it cannot ensure its suitability for a realistic mobile-phone environment. For this reason, our protocol does not have any additional algorithms for achieve suitability on realistic mobile-phone environment. Moreover, some papers noted a defect in the UMTS AKA protocol with regards to the plain code transmission between the SN and the HE. Since the SN and the HE are immovable equipment and linked by wires, eavesdropping and distortion of their signals does not occur easily and they also have enough computing power to cope with the needs of encryption. Therefore, our protocol does not particularly illustrate the security program of this block. In fact,

Table 3: The Comparison of Communication Overhead

	UMTS-AKA	ER-AKA	AP-AKA	X-AKA	Our proposed
Network Traffic on the Phase 1 (Distributed of authentication vector(Proxy key) Phase))					
SN→MS	None	None	128	None	None
MS→SN	128	192+Size(Chain+TS)	192	320	256
SN→HE	128	192+Size(Chain+TS)	320	320	256
HE→SN	560×n	560+Size(Chain)	608×n	368	256
SN→MS	None	192+Size(Chain)	None	None	None
Subtotal	256+560×n	1136+Size(Chain+TS)×2	640+608×n	1008	768
Network Traffic on the Phase 2 (Authentication and Key Establishment Phase)					
SN→MS	240	Size(Hash)	320	368	192
MS→SN	64	Size(Hash)	64	64	192
Subtotal	304	Size(Hash)×2	384	432	384

any cryptography mechanism can work out the security problem of this block.

This paper proposes an enhanced protocol on the UMTS AKA protocol. We use the similar technique to strengthen the original protocol and yield good results. We also repaired the defects of the UMTS AKA protocol such as reduction of storage data, enhancement of mutual authentication, and solving the program with the sequence number which was not easily maintained. We also compared our proposed with the other protocol such as UMTS AKA, ER-AKA, AP-AKA, and X-AKA on the computational overhead, storage, and communication overhead, and proved that our proposed excellence. Therefore, our proposed not only outstanding but also can be immediately adopted in the current environment.

At present, the future of 4G protocols has been proposed, such as 3GPP LTE, WiMAX. We remain committed to the continuation of the AKA protocol to enable him to meet the environmental needs of the future 4G.

References

[1] 3GPP, "3rd generation partnership project, technical specification group services and systems aspects, 3G security, specification of the milenage algorithm set, document 1: General," tech. rep., 3GPP TS 35.205 V6.0.0 (2004-12).

[2] 3GPP, "3rd generation partnership project, technical specification group services and systems aspects, 3G security, specification of the milenage algorithm set, document 2: Algorithm specification," tech. rep., 3GPP TS 35.206 V6.0.0 (2004-12).

[3] 3GPP, "3rd generation partnership project, technical specification group services and systems aspects, 3G security, specification of the milenage algorithm set, document 5: Summary and results of design and evaluation," tech. rep., 3GPP TS 35.909 V6.0.0 (2004-12).

[4] 3GPP, "3rd generation partnership project, technical specification group services and systems aspects, 3G security, security architecture," tech. rep., 3GPP TS 33.102 V7.1.0 (2006-12).

[5] Hassan Aljifri and Nizar Tyrewalla, "Security model for intra-domain mobility management protocol," *International Journal of Mobile Communications*, vol. 2, pp. 157–170, 2004.

[6] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Tran. Computer Systems*, vol. 8, no. 1, pp. 18–36, 1990.

[7] Ting-Yi Chang and Min-Shiang Hwang, "User-Anonymous and Short-Term Conference Key Distribution System via Link-Layer Routing in Mobile Communications," *International Journal of Mobile Communications*, vol. 9, pp. 144–158, 2011.

[8] Christos K. Dimitriadis and Siraj A. Shaikh, "A Biometric Authentication Protocol for 3G Mobile Systems: Modelled and Validated Using CSP and Rank Functions," *International Journal of Network Security*, vol. 5, no. 1, pp. 99-111, 2007.

[9] N. El-Fishway and A. Tadros, "On the design of authentication protocols for third generation mobile communication systems," in *Proceedings of the Twentieth National Radio Science Conference*, pp. C24_1–C24_10, March 2003.

[10] Kalid Elmufti, Dasun Weerasinghe, Muttukrishnan Rajarajan, and Veselin Rakocevic, "Anonymous authentication for mobile single sign-on to protect user privacy," *International Journal of Mobile Communications*, vol. 6, pp. 760–769, 2008.

[11] E. T. S. I. (ETSI), "Recommendation GSM 03.20, security related network functions," tech. rep., June 1993.

[12] Vincenzo Falletta and Fabio Ricciato, "Detecting Scanners: Empirical Assessment on a 3G Network," *International Journal of Network Security*, vol. 9, no. 2, pp. 143-155, 2009.

[13] C. F. Grecas, S. I. Maniatis, and I. S. Venieris, "Towards the introduction of the asymmetric cryptography in GSM, GPRS, and UMTS networks," in *Proceedings of the Sixth IEEE Symposium on Computers and Communications*, pp. 15–21, July 2001.

[14] O. Group, "Architecture for public-key infrastructure," *Open Group Draft*, May 1997.

[15] L. Harn and W.-J. Hsin, "On the security of wireless network access with enhancements," in *Proceedings of the 2003 ACM workshop on Wireless security*, (San Diego, CA, USA), pp. 88–95, 2003.

[16] Y. Zhi Hu, D. wei Ma, and X. fei Li, "An improved authentication protocol with less delay for UMTS mobile networks," in *Proceedings of the International Conference on Networking and Digital Society (ICNDS '09)*, vol. 2, pp. 111 – 115, 2009.

[17] C.-M. Huang and J.-W. Li, "Authentication and key agreement protocol for UMTS with low bandwidth consumption," in *Proceedings of 19th International Conference on Advanced Information Networking and Applications (AINA 2005)*, vol. 1, pp. 392–397, March 2005.

[18] Wen-Shenq Juang and Jing-Lin Wu, "Robust and efficient authenticated key agreement in mobile communications," *International Journal of Mobile Communications*, vol. 7, pp. 562–579, 2009.

[19] Cheng-Chi Lee, Min-Shiang Hwang, I-En Liao, "A New Authentication Protocol Based on Pointer Forwarding for Mobile Communications," *Wireless Communications & Mobile Computing*, vol. 8, pp. 661–672, 2008.

[20] Cheng-Chi Lee, I-En Liao, Min-Shiang Hwang, "An Extended Certificate-based Authentication and Security Protocol for Mobile Networks," *Information Technology and Control*, vol. 38, pp. 61–66, 2009.

[21] Chun-Ta Li, Chou-Chen Yang, and Min-Shiang Hwang, "A Secure Routing Protocol with Node Selfishness Resistance in MANETs," to be appear in *International Journal of Mobile Communications*.

[22] Jung-Wen Lo, Cheng-Chi Lee, Min-Shiang Hwang, Yen-Ping Chu, "A Secure and Efficient ECC-based AKA Protocol for Wireless Mobile Communications," *International Journal of Innovative Computing, Information and Control*, vol. 6, pp. 5249–5258, 2010.

[23] M. Naor and M. Yung, "Universal one-way hash functions and their cryptographic applications," in *Proceedings of the twenty-first Annual ACM Symposium on Theory of Computing, Seattle, Washington*, pp. 33–43, 1989.

[24] Hsia-Hung Ou, Min-Shiang Hwang and Jinn-Ke Jan, "The UMTS-AKA Protocols for Intelligent Transportation Systems," *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, pp. 1-12, 2009.

[25] Hsia-Hung Ou, Min-Shiang Hwang and Jinn-Ke Jan, "A provable billing protocol on the current UMTS," *Wireless Personal Communications*, vol. 55, pp. 551-556, 2010.

[26] Hsia-Hung Ou, Min-Shiang Hwang and Jinn-Ke Jan, "A Cocktail Protocol with the Authentication and Key Agreement on the UMTS," *Journal of Systems and Software*, vol. 83, pp. 316-325, 2010.

[27] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems," *Communications of the ACM*, vol. 21, pp. 120–126, Feb. 1978.

[28] Babu B. Sathish and Pallapa Venkataram, "A Dynamic Authentication Scheme for Mobile Transactions," *International Journal of Network Security*, vol. 8, no. 1, pp. 59-74, 2009.

[29] K. Shim, "Cryptanalysis of mutual authentication and key exchange for low power wireless communications," *IEEE Communications Letters*, vol. 7, pp. 248–250, May 2003.

[30] Chou-Chen Yang, Kuan-Hao Chu, and Ya-Wen Yang, "3G and WLAN Interworking Security: Current Status and Key," *International Journal of Network Security*, vol. 2, no. 1, pp. 1-13, 2006.

[31] M. Zhang and Y. Fang, "Security analysis and enhancements of 3GPP authentication and key agreement protocol," *IEEE Transactions on Wireless Communications*, vol. 4, pp. 734–742, March 2005.

Appendix

The Formal Security Analysis

In order to prove that our enhanced UMTS AKA protocol can reach the goals of authentication and key agreement. The formal method BAN logic [6] is utilized to analyze in this section. BAN logic is one of the authentication logics to analysis cryptographic protocols, authentication protocol especially. It was used to analyze a wide range of the authentication protocol. The formal analysis of our enhanced UMTS AKA protocol is as follows:

1. The proposed protocol:

   - (Message 1) $MS \rightarrow SN \rightarrow HE : IMSI, Seed$;
   - (Message 2) $HE \rightarrow SN : IMSI, PK$;
   - (Message 3) $SN \rightarrow MS : RES_1, RAND_2$;
   - (Message 4) $MS \rightarrow SN : RES_2, RAND_1'$.

   The relation parameters ale listed in the following:

   (a) $Seed$: A random number.

   (b) PK: $f4_K(Seed)$; K is a secure key which is shared between the MS's USIM and the MS's HE. The $f2, f3$ and $f4$ are the key generating functions.

   (c) RES_1: $f2_{PK}(RAND_1)$; $RAND_1$ is a random number which is choose and deliver on last round by MS. If this is the authentication of the first time, then $RAND_1 = Seed$.

   (d) $RAND_2$: A random number.

   (e) RES_2: $f2_{PK}(RAND_2)$.

   (f) $RAND_1'$: A random number which is use at the authentication of next time.

   (g) CK: $f3_{PK}(RAND_2)$; A cipher key.

   (h) IK: $f4_{PK}(RAND_2)$; A Integrity Key.

2. Security Assumptions:

   (a) It is assumed that K is a secure key which is share between the MS and his HE.

   (a) MS has the secure key K;

   (b) HE has the secure key K;

   (c) MS$|\equiv$ MS$\overset{K}{\rightleftarrows}$ HE;

   (d) HE$|\equiv$ MS$\overset{K}{\rightleftarrows}$ HE.

   (b) It is assumed that the trusting relationship between HE and SN.

(a) SN$|\equiv$ HE$|\Rightarrow$ MS$\overset{K}{\rightleftarrows}$ HE.

(c) It is assumed that the communication between HE and SN is secure.

(a) SN$|\equiv$ SN$\overset{P}{\rightleftarrows}$ HE, P is the conveyance messages between SN and HE.

(b) HE$|\equiv$ SN$\overset{P}{\rightleftarrows}$ HE, P is the conveyance messages between SN and HE.

3. Formally messages:

(a) (Message 1) $MS \rightarrow SN \rightarrow HE : IMSI, Seed$;

(b) (Message 2) $HE \rightarrow SN : IMSI, f4(K, Seed)$;

(c) (Message 3) $SN \rightarrow MS : f2(f4(K, Seed), RAND_1), RAND_2$;

(d) (Message 4) $MS \rightarrow SN : f2(f4(K, Seed), RAND_2), RAND'_1$.

4. Protocol goals:

(a) Mutual authentication between MS and SN.

(b) Key agreement between MS and SN.

(c) Key freshness between MS and SN.

(d) Confidentiality between MS and SN.

5. Statements and analysis:

(a) (Goal 4.a) Mutual authentication between MS and SN.

(1) (3.a)(2.c.2)$\rightarrow MS \mid\equiv \sharp(Seed) \wedge SN \mid\equiv \sharp(Seed) \wedge HE \mid\equiv \sharp(Seed)$;

(2) Since (2.a.2), (2.a.4), (2.b.1), (2.c.1). By (3.b)$\rightarrow SN \mid\equiv \forall f4(K, Seed).(HE \mid\Rightarrow SN \overset{f4(K,Seed)}{\leftrightarrow} MS)$;

(3) Since (2.a.1), (2.a.3) and (5.a.1)$\rightarrow MS \mid\equiv MS \overset{f4(K,Seed)}{\leftrightarrow} HE$;

(4) For message-meaning rule and (3.c):

$$\frac{MS \mid\equiv MS \overset{f4(K,Seed)}{\leftrightarrow} HE, MS \lhd f4(f4(K, Seed), RAND_1}{MS \mid\equiv HE \mid\sim f4(f4(K, Seed), RAND_1}$$

(5) For nonce-verification rule, (1.c) and (5.a.4)

$$\frac{MS \mid\equiv \sharp(RAND_1), MS \mid\equiv HE \mid\sim f4(f4(K, Seed), RAND_1}{MS \mid\equiv HE \mid\equiv f4(f4(K, Seed), RAND_1)}$$

(6) For jurisdiction rule and (5.a.5)

$$\frac{MS|\equiv HE|\Rightarrow f4(f4(K,Seed),RAND_1),MS\lhd SN|\sim f4(f4(K,Seed),(RAND_1)}{MS \mid\equiv HE \mid\equiv SN}$$

(7) For message-meaning rule (5.a.2) and (3.d)

$$\frac{SN \mid\equiv SN \overset{f4(K,Seed)}{\leftrightarrow} MS, SN \lhd f4(f4(K, Seed), RAND_2}{SN \mid\equiv MS \mid\sim f2(f4(K, Seed), RAND_2}$$

(8) For nonce-verification rule, (1.b), (1.d), (1.e) and (5.a.7)

$$\frac{SN \mid\equiv \sharp(RAND_2), SN \mid\equiv MS \mid\sim f2(f4(K, Seed), RAND_2}{SN \mid\equiv MS \mid\equiv f2(f4(K, Seed), RAND_2)}$$

- **(9)** For jurisdiction rule and (5.a.8)
  $$\frac{SN|\equiv MS|\Rightarrow f2(f4(K,Seed),RAND_2),SN\triangleleft MS|\sim f2(f4(K,Seed),(RAND_2}{SN \mid\equiv MS}$$
- **(10)** By (5.a.6), (5.a.9) $\rightarrow MS \mid\equiv HE \mid\equiv SN \wedge SN \mid\equiv MS \rightarrow MS \mid\equiv SN \wedge SN \mid\equiv MS$, So, the goal of mutual authentication between MS and SN is holds.

(b) (Goal 4.b) Key agreement between MS and SN.

- **(1)** There are two keys to agreement between MS and SN: CK and IK.
- **(2)** (1.g)$\rightarrow CK = f3(PK, RAND_2) = f3(f4(K, Seed), RAND_2)$
- **(3)** Since (5.a.1), (5.a.2), (5.a.3)$\rightarrow SN \mid\equiv (SN \stackrel{f4(K,Seed)}{\leftrightarrow} MS) \wedge MS \mid\equiv (SN \stackrel{f4(K,Seed)}{\leftrightarrow} MS)$
- **(4)** (3.c)$\rightarrow SN \mid\equiv (SN \mid\sim (RAND_2) \wedge \sharp(RAND_2))$
- **(5)** (3.d), (5.a.8)$\rightarrow SN \mid\equiv (MS \mid\equiv \sharp(RAND_2))$
- **(6)** (3.c)$\rightarrow MS \mid\equiv (MS \triangleleft RAND_2 \wedge SN \mid\sim RAND_2 \wedge \sharp(RAND_2))$
- **(7)** By (5.b.3), (5.b.5), (5.b.6) $\rightarrow$ CK agreement between MS and SN
- **(8)** (1.h)$\rightarrow IK = f4(PK, RAND_2) = f4(f4(K, Seed), RAND_2)$
- **(9)** As (5.b.7)$\rightarrow$ IK agreement between MS and SN.
- **(10)** By (5.b.7), (5.b.9)$\rightarrow$ the goal of key agreement between MS and SN is holds.

(c) (Goal 4.c) Key freshness between MS and SN.

- **(1)** Since (5.a.1), (5.b.3), (5.b.4), (5.b.6)$\rightarrow MS \mid\equiv (\sharp(PK) \wedge \sharp(CK) \wedge \sharp(IK))$. $\rightarrow$ So, the goal of key freshness between MS and SN is hold.

(d) (Goal 4.d) Confidentiality between MS and SN.

- **(1)** Since (5.a), (5.b), (5.c), for message-meaning rule$\rightarrow$
  $$\frac{MS|\equiv MS(\stackrel{CK\wedge IK}{\leftrightarrow}SN),MS\triangleleft(Message)_{(CK\wedge IK)}}{MS|\equiv SN|\sim Message} \wedge \frac{SN|\equiv MS(\stackrel{CK\wedge IK}{\leftrightarrow}SN),SN\triangleleft(Message)_{(CK\wedge IK)}}{SN|\equiv MS|\sim Message}$$
- **(2)** By (5.d.1)$\rightarrow$ the goal of confidentiality between MS and SN is hold.

Since (5.a.10), (5.b.10), (5.c.1), and (5.d.2), all the protocol goals are hold.