A provable billing protocol on the current UMTS *

Hsia-Hung Ou‡ Min-Shiang Hwang† Jinn-Ke Jan‡

Department of Management Information Systems †
National Chung Hsing University
250, Kuo Kuong Road,
Taichung, Taiwan 402, R.O.C.
Email: mshwang@nchu.edu.tw

Department of Computer Science and Engineering ‡
National Chung Hsing University
Taichung, Taiwan 402, R.O.C.

September 8, 2012

*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: Prof. Min-Shiang Hwang.

A provable billing protocol on the current UMTS

Abstract

This paper proposes a simple method that equips UMTS-based telecom companies with a mechanism to prove the records on mobile users' phone bills. In the history of mobile phone communication, we have seen countless unsettled disputes where the mobile user disagrees with the telecom company either on the calling time or on the duration, or even on whether or not a call was actually made. In this paper, a provable billing protocol will be presented that can effectively solve disagreements between the two parties. Equipped with a non-repudiation function, the proposed protocol enables the service provider to hold on to the solid proofs as to exactly when and to which number a mobile phone user made a call so that the mobile user cannot deny; at the same time, the mobile user also gets to have his/her own share of proofs as to when and how the mobile services were accessed, so that the bill can be double checked to make sure nothing goes wrong. And, to make it even better, this new protocol is perfectly compatible with the standard UMTS protocol and is therefore readily applicable to the current mobile phone communication environments.

Key Words: UMTS, Billing, Charging. AKA, Mobile Communication System

1 Introduction

Mobile communication has gone an incredibly long way since the debut it made just decades ago. At the present time, in addition to the main stream second generation system, namely GSM (Global System for Mobile Communication) [16], the third generation mobile telecommunication system, UMTS (Universal Mobile Telecommunication System) [37], has also reached maturity and begun its service, gaining great popularity and becoming a red hot choice for subscribers. In comparison with GSM, UMTS enjoys the advantage of a greater bandwidth and is therefore capable of offering a number of additional kinds of services besides the ordinary voice call and voice message. Such power and convenience the third generation system brings of course mean more business and greater profits to the mobile telecommunication company.

Accompanying the variety of services now available to mobile telecom system subscribers, there comes the issue of billing. Viewed from the angle of the mobile telecom company, an ideal billing system is one that can take down exact, detailed records both as proofs of the subscriber's accesses to the services and as information for correct billing. However, from the viewpoint of the subscribers, a system can by no means be called perfect if they are not provided with a mobile device programmed to be able to take down all those detailed records on their own side real time which can later be raised as proofs when the mobile service users doubt the credibility of the bills they receive from the company. Lack of such a double-checking, disagreement-avoiding design is in fact the biggest problem mobile communication systems nowadays share in common. Instead, protocols presently in use are mostly system-provider-dominant, leaving no room for the users to even raise their doubts about the correctness of the bills. As a result, so many disputes just go unsolved. To shut the door on such billing problems with both the service provider and the customers satisfied, in this paper, we shall dig deep into the billing protocol of the more and more prevalent UMTS design and find out where the roots of these problems are. We will present our scheme to solve the billing trouble in next section.

1.1 The Current UMTS Charging System

UMTS is quite a complex system because it is its job to integrate many resources and provide a wide variety of services. For example, there have to be both circuit-switched and packet-switched links; at the same time, compatibility problems also

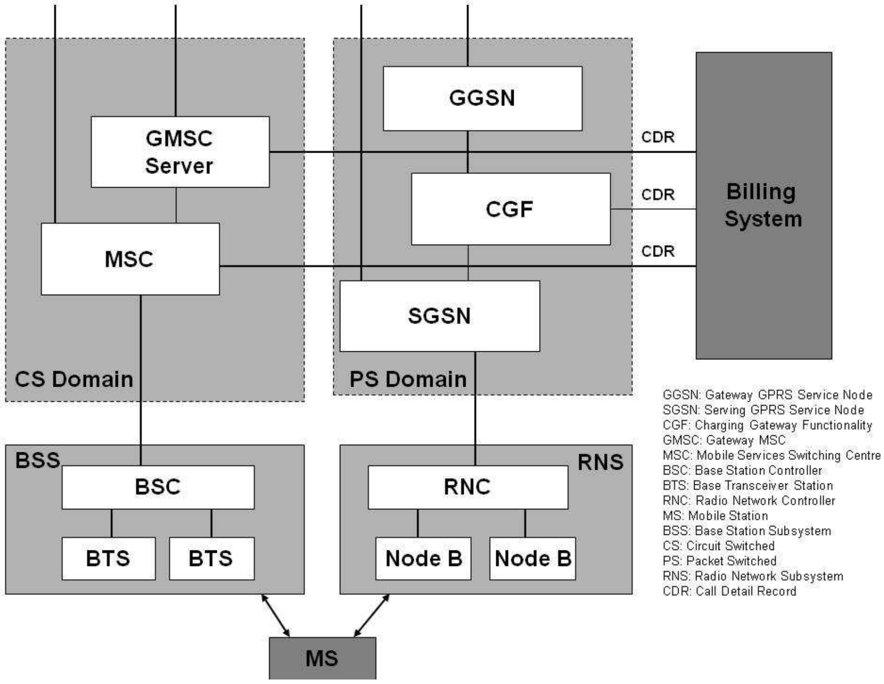

Figure 1: UMTS Charging Logical Architecture

have to be taken care of because UMTS has to cover older, lower grade systems such as GSM. So, there is supposed to be some complexity in the UMTS billing system. Simply put, the current UMTS billing system works by taking down individual subscribers' service accessing records, namely the CDR (charging data record). According to the CDR, the UMTS billing system can figure out the charges and add them up. The charging logical architecture of UMTS is shown in Figure 1 [3]. There are two domain networks, PS and CS domain, for MS to charge. When the MS uses in CS domain network, the MSC (Mobile Switching Centre) and GMSC (Gateway MSC) are responsible for the record of all charging relevant information and deliver the information to the bill system. When the MS uses in PS domain network, the CDR which is generated by SGSN (Serving GPRS Support Node) and GGSN (Gateway GSN) is forwarded via the CGF (Charging Gateway Function) to the Billing System.

CDR comes in various forms, depending on what it is used for. Some CDR formats do not necessarily have to do with charging. For example, it can be used solely to keep track of subscribers' behaviors and message exchanges [3]. In fact, charging-related CDR can take different shapes, depending on the characteristics of the charged entity [1] and the environment it is in [9]. For example, subscribers will offer different CDR information from that provided by service providers, and by the same token the CDR details used by the circuit switched domain [7] will not be the same as those used by the packet-switched domain [8]. However, roughly speaking, the information CDR keeps is right what is needed for charging and billing. The details include the subscriber's identity, services accessed, time and duration of service access, quantities of resources utilized, etc. These details help the billing system figure out the charges to put down on the bill. As for the transmission of these CDR details, they follow internal routes to the billing system [10]. An access to the communication services can mean the generation of with one CDR or more than one. In the meantime, the charging data can be delivered to the billing system by means of either online charging or offline charging [2], and the charging information can be sent out right after the completion of a communication service, or the bill can reach after every regular time interval, or of course a bill can also come to charge for all the communication services accessed during a relatively long period of time.

1.2 Review of the Literature

The literature concerned is mostly focused on the authentication part because it is the foundation of charging and billing. Of course no charging can be done right without proper confirmation of the identities of the subscribers. Therefore, most UMTS-related researches that do mention the subject of billing [11, 15, 19, 27, 29, 32, 33, 36] deal with authentication and non-repudiation problems. These schemes endeavor to enable system providers to identify users so that nobody can pretend to be a legal user and gain access to the services without subscribing first. Besides that, once a subscriber used any of the services, there is no way he/she can deny it. However, such schemes have no power in confirming the types and quantities of the services consumed. Although they mainly concern CDR, it is the authentication part that they put emphasis on instead of the contents of CDR. In addition, many researches put their stress on the construction of a whole framework and the integration of the existing systems [17, 24, 25, 28, 39]. Their aim is to build up a charging module independent of the existing systems that can take care of all the billing-related affairs. Among the researches, some takes the charging module for some kind of middleware. Such methods are usually developed to fit applications not only with UMTS but also other wireless network systems like GSM, wireless Lan, Bluetooth, and so on. Of course, there are also some other extensions to make from these methods. For example, SET (Secure Electronic Transaction Protocol) is a mechanism that can be taken in as a mobile charging standard [22]. In addition, some researchers suggest that charging and billing can be viewed from the angle of payment [18, 23, 31, 34]. New technologies such as E-cash (brought up in [23]) and Electron Tickets (mentioned in [35]) can also help. However, a comparatively big difference is made in [13], where concepts of Hash Chain and KryptoKnight cryptography [12, 20, 21] are taken in to enable their protocol to reach non-repudiation without using PKI.

The research efforts mentioned above have each made its own contribution to the charging and billing problem solving. Of course, each has its own strengths and weaknesses. However, none of the above methods addresses the biggest problem the current UMTS billing system design has: that the service system provider plays a super dominant role while the service subscribers can do nothing but accept in submission whatever the bill says. When a subscriber does not agree with a certain payment listed in the bill, unless some proof outside the system can be raised, which is usually either impossible or not worth all the trouble, it is simply no use for the subscriber to argue with the service provider because the system is completely in the service provider's favor. Of course such a design is very unfair to the consumer. Even the non-repudiation mechanism mentioned above is designed for the service provider to be able to shut a disagreeing subscriber's mouth instead of making sure that the rights on both sides are properly preserved.

To make a difference, Chen et al. [14] proposed a secure, fair billing system for the use of GSM. Due to the low computation power of the mobile device, their system uses an additional piece of equipment, namely the server by the name of Observer, to enable the subscriber to check out and sign on the charges the service provider raises. This way, no payment can be asked for and will be paid without the consent from both sides and no dispute will ever happen again. They charge according to the timestamps recorded. At the beginning and the end of every communication, from the subscriber's side, a hash chain will be sent to the service provider as proof, and then the service provider will send the proof along with its signature on the current time to the Observer. If the Observer checks out the data sent in and the result is positive, then it signs on behalf of the subscriber and sends the records back to the service provider. This way, by signing for the subscriber, the Observer achieves non-repudiation for both sides, and in case there is a dispute in the future, those signatures can be used to confirm the time and duration of each phone call. However, in spite of this great advancement Chen et al. have made, there are in fact some minor weaknesses in their protocol. That are, Synchronization program, Additional offline job required and storage space required.

In 2008, Li et al. [26] pointed out the Chen et al.'s proposed is vulnerabilities in two aspects. Their proposed is not real fair and synchronization program. Afterward Li et al. proposed an undeniable mobile billing scheme. However, their scheme did not integrate with the authentication protocol.

Recently, the authors have proposed a method to settle charging disputes that occur when the mobile user roams across network domains run by different communication service providers [30]. Instead of using time records as proof for charging like what is done in many traditional methods, in the protocol, we use hash chains to represent the communication time, one hash value proving one time unit (e.g. a minute) of communication consumption. During mobile communication, the subscriber's mobile device sends one hash value after another to the service provider, and the service provider keeps the last hash value as proof. This way, disagreements can be settled among cooperating mobile communication service providers. In this paper, the authors wish to offer a new, integrated and practical protocol built up on the basis of their previous endeavors that can solve all the charging and billing problems between the subscriber and the telecommunication service provider.

The major difference between the new protocol this paper presents and its predecessors is that the new protocol keeps UMTS the way it is without making any modification. Since UMTS is already a popular system prevalently used, any modification can mean serious damage to its practicability. In real applications, the new protocol can be attached to the original UMTS structure that has been running for years and make the whole system more powerful without causing any serious compatibility problems. The aim of this paper is to explore the possibility for UMTS to be strengthened and become a fair system both to the service provider and to the subscriber when it comes to charging and billing. With the rights on both sides properly preserved, the subscriber will no longer have to live with the fact that every penny has to be paid the way the bill says just because it says so; instead, it is the service provider's job to prove to the subscriber that every line on the bill actually comes from some consumption behavior signed and recorded by both parties. In the meantime, the new protocol can also rule out the possibility of a subscriber cheating by denying accessing the service because everything that has been done leaves a signed record and there is simply no denying.

2 The Proposed Scheme

The UMTS AKA (Authentication and Key Agreement) protocol [4] is constructed by 3GPP and has been prevalently used in UMTS-based systems everywhere. The security of the UMTS AKA protocol mainly relies on a secret key K stored in the USIM (Universal Subscriber Identity Module) card of the subscriber's mobile device (or MS, meaning Mobile Station). This secret key K is only known to the USIM

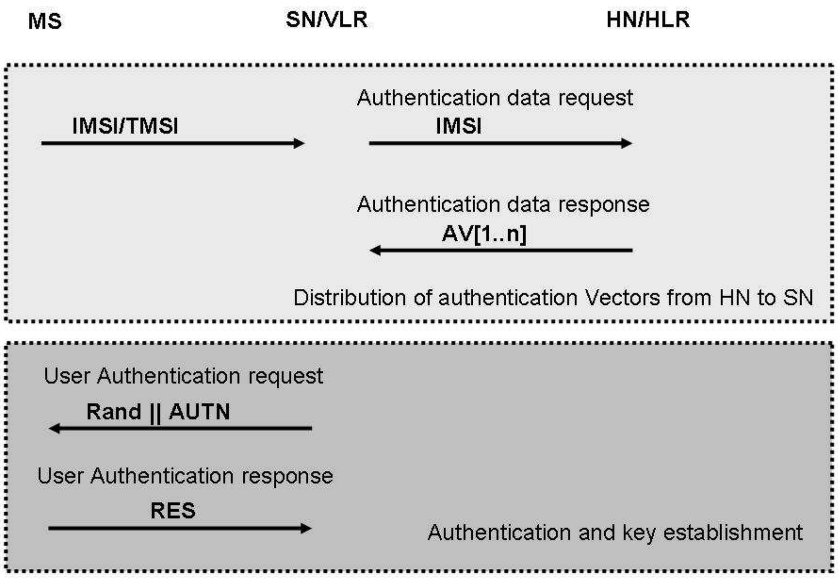

Figure 2: UMTS AKA Protocol

and the telecommunication service provider (HN, Home Network) to the subscriber. With this key in hand, wherever it is, the subscriber can do authentication and get connected with the local service network (SN) through the UMTS AKA protocol. Figure 2 is an illustration of how the UMTS AKA protocol works.

UMTS was first put to use many years ago and has been operating well since. However, one problem with UMTS has remained unsolved. When the subscriber and the service provider disagree on the figures printed out on the check, it is always the subscriber, namely the consumer, that reluctantly takes the disadvantaged side. In fact, unfair though it seems, practically the bill is always right, and the consumer can pretty much do nothing about it unless some solid evidence can be raised to prove that the bill is wrong, which is usually extremely difficult and troublesome. The telecom company, who takes the upper hand over this matter, does have a big chance to mess with the figures on the bill and go unnoticed. In an attempt to solve this problem, in this section we shall present a new charging and billing mechanism we have designed. Our hopes are that the new protocol can function well in the

UMTS environment and that it can satisfy the following requirements:

1. It should be able to solve billing disputes evenly, allowing both the service provider and the consumer to have a fair shot in proving their point. In other words, the new mechanism should provide UMTS with non-repudiation.

2. It should fit every UMTS operation, making as little modifications as possible.

3. It should be able to deal with all the ways of payment present in the current UMTS environment.

2.1 Foundation

In our new protocol, hash chains are used as tokens for payments because of their irreversibility. Every hash value stands for an equal amount of money. For example, suppose a subscriber has a_0. From a_0 the following hash chain $\{a_1, a_2, \cdots, a_i\}$ can easily be derived through the hash function. When it is time to pay the bill, the payments will be done in reverse order starting with a_i, namely $(a_i, a_{i-1}, a_{i-2}, \cdots, a_1)$. Due to the irreversibility of the hash function, the service provider cannot derive any a_{n-1} from a_n. Therefore, all the service provider has to show is the last hash value, and then the accounts can be settled and proven for the subscriber. In our protocol design, every hash value represents a fixed unit. For example, if the service the SN provides is supposed to be paid for by the minute, then the MS will deliver one hash value to the SN every minute, and the last hash value the SN collects reveals the total time of service the MS has consumed. Sometimes, however, instead of the minute or the second, the charge that the SN asks for is counted by the times of service. For example, suppose it costs one X dollars to download a piece of music for the phone ring. If every hash value stands for Y dollars, then the MS will send as many as X/Y hash values to the SN when a piece of music has been downloaded. In this case, suppose the first hash value in the chain to be used on account of this music download is a_j. If $X/Y = k$, then all the MS has to send to the SN is a_{j-k+1} because it is enough to show that k units, namely Y dollars, have been consumed.

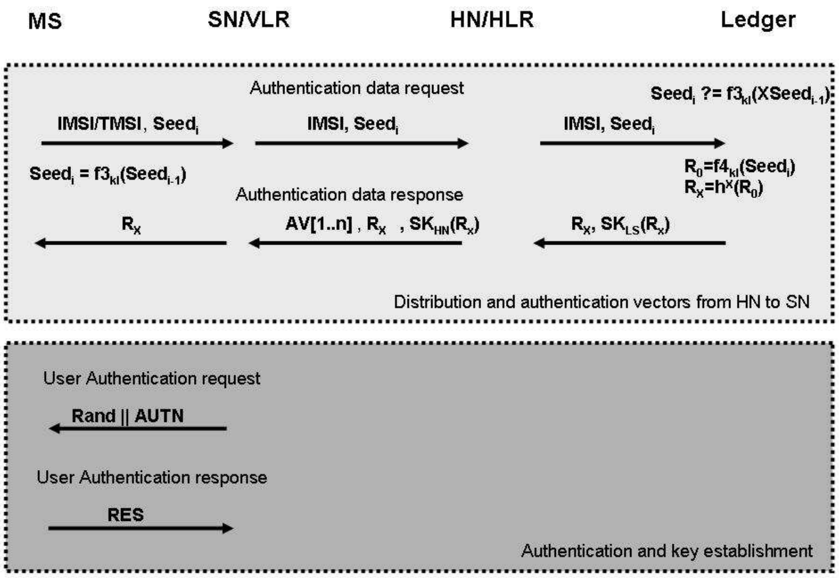

Figure 3: Our AKA charging protocol

2.2 Our AKA charging protocol

To satisfy the requirement of non-repudiation, it is probably the best way to bring in the digital signature. However, due to the low computation power of the mobile devices nowadays, it seems impractical to even think about the mobile phone holder digitally signing anything. To make things worse, the USIM card in the mobile phone does not have a big enough storage space so that the subscriber's records of telecom service accesses can be saved in the mobile phone. To get around these obstacles, we bring in the concept of assistant server. We call the assistant server employed by our new protocol the ledger server. To both the subscriber and the service provider, the ledger server is a third party worthy of the trust from both. The subscriber can trust the ledger server with his/her digital signature and signed records.

As we mentioned earlier, to make sure our proposed protocol is compatible with UMTS, we build our structure upon the UMTS AKA protocol. Figure 3 shows the detailed framework of our new protocol.

Just like the original UMTS AKA protocol, our AKA charging protocol is composed of two parts, namely the authentication vector distribution phase and the authentication and key establishment phase. The major difference between the original AKA protocol and our AKA charging protocol, however, is that our new protocol has a new party involved, namely the ledger server, with some additional parameters attached to the ends of the original authentication parameters. The attached parameters help ensure the compatibility of the our protocol with the original UMTS design. The telecom company can either leave out the additional parameters and offer the original services only or bring in the additional parameters and offer not only the original services but also the extended services. At the same time, the subscriber also can choose either to leave out the additional parameters and ask for only the original services or to bring in the additional parameters and enjoy both the original services and the extended services. This way, the original AKA can function properly with or without the existence of the our protocol.

Here are some steps to take in our authentication vector distribution phase:

1. The MS provides not only its $IMSI/TMSI$ for identification but also a seed ($Seed_i$) to the SN. Then, the SN forwards the received message to the MS and the HN like it does in the original UMTS AKA protocol. What is different is that the HN will forward the message it receives from the SN to the ledger server. In the message produced by the MS, there is $Seed_i = f3_{kl}(Seed_{i-1})$, $Seed_{i-1}$ stands for the seed used for authentication last time, and the original $Seed_0$ and secret key kl are decided by both the MS and the ledger and only known to them both. In addition, $f3$ is the key generating function of the standard UMTS [5, 6]. We use it to produce the association between the seeds we use at adjacent times that makes sure that the $Seed_i$ the ledger server receives was actually sent out by the MS, not by the SN or the HN. This can also be considered as a kind of authentication too. Therefore, the ledger has to keep the $Seed_i$ the MS used last time, namely $XSeed_{i-1}$, and confirm

$Seed_i = f3_{kl}(XSeed_{i-1})$ for the $Seed_i$ received from the HN. If the result of the confirmation turns out negative, the ledger server will refuse to offer any endorsement.

2. After receiving the message, the ledger server checks the IMSI and takes out the secret key kl that the ledge server shares with the MS corresponding to the $IMSI$ in advance. Then the initial value of the hash chain is computed as $R_0 = f4_{kl}(Seed_i)$, where $f4$ is also a key generating function in the standard UMTS system [5, 6] just like $f3$. After generating R_0, the ledger server hashes it X times to get $R_X = h^X(R_0)$. Then, the ledger server uses its own private key to sign R_X, and the result is $SK_{LS}(R_X)$. Now R_X and $SK_{LS}(R_X)$ are sent back to the HN.

3. Upon receiving the message from the ledger server, the HN uses the ledger server's public key to check if $SK_{LS}(R_X)$ is correct. If everything is fine, the HN stores the received R_X and $SK_{LS}(R_X)$ for later use as proof. Then, the HN uses its own private key to sign R_X, and the result is $SK_{HN}(R_X)$. Now the HN can use the secret key k it shares with the MS to derive the authentication vector AV required by the original AKA protocol. Finally, the HN sends back R_X, $SK_{HN}(R_X)$ and AV to the SN. Upon receiving the message from the HN, the SN uses the public key of the HN to check whether the received $SK_{HN}(R_X)$ is correct. If so, it stores the received R_X and $SK_{HN}(R_X)$ for later use as proof. In this place, please note that what the HN receives is $SK_{LS}(R_X)$ signed by the ledger server, while what the SN receives is $SK_{HN}(R_X)$ signed by the HN. The reason for that is the possibility of charging relationships occurring between the SN and the HN [30].

4. Finally, the SN sends the R_X it receives back to the MS so that the MS can confirm that the ledger server has accessed the data concerned and that no mistakes have been going on. By doing this, we can make sure that no

synchronization problems happen to the seed $Seed_i$ shared between the MS and the ledger server due to information loss. In addition, to take further precautions, both the MS and the ledger server will keep the seeds used the previous time and the time before the previous time, namely $Seed_{i-1}$ and $Seed_{i-2}$. If the $Seed_{i-1}$ values do not agree, then $Seed_{i-2}$ will be used to make sure no synchronization problems occur in our protocol.

The second phase, namely the authentication and key establishment phase, of our AKA charging protocol is the same as that of the original UMTS AKA protocol. It goes as follows:

1. The SN extracts the first fresh string of parameters from the AV of the MS received from the HN. Then the SN sends the random number *Rand* and the authentication parameter $AUTN$, both taken out from the string of parameters extracted from the AV, to the MS.

2. Upon receiving the *Rand*, $AUTN$, and R_X from the SN, the MS checks the correctness of these data to make sure of the identity of the SN. Then, by using the Rand and the secret key it shares with the HN, The MS figures out the RES and some other data to be used later for the connection such as key IK, and CK. Now the MS sends the RES back to the SN. Note that the purpose of the $R_X$checkout is to make sure that neither the SN nor the HN is cheating.

3. The SN compares the received RES with the one taken out from the AV. If the result is positive, then the MS proves to be the lawful subscriber. Now the CK and IK from the AV can be used to create a connection.

Authentication is the basis of all charging systems. The original UMTS AKA protocol can already do perfect authentication and identify the subscriber. Then, the extra parameters we add to the protocol can help take care of the charging and billing problems that have long been unsolved. The next subsection will specify how those problems are dealt with.

2.3 Charging, Billing and Dispute

Figure 4 is an illustration of our charging protocol. As we elaborated earlier, the hash chain is what we use as the certificate for charging. As the authentication process detailed above goes, when the subscriber reaches the territory of a new SN, the registration has to be done first. In our protocol, the subscriber will start by sending a *seed* to the SN. By way of the HN, the ledger server will also obtain this *seed* and figure out R_0 accordingly. Then, R_0 gets hashed X times and becomes $R_X = h^X(R_0)$, which is then sent out to the HN and the SN. On the other hand, since the MS also has R_0, the MS too has the ability to come by $R_1, R_2, \cdots, R_X$ in such order. As for the SN and the HN, only R_X is available, which of course does not make it possible for them to derive $R_0, R_1, R_2, \cdots, R_{X-1}$ due to the irreversibility of the hash function. Each charge unit's value is decision in the authentication phase. The Charge unit is possibly the time or packet which is decided by the service type. For example, if a certain kind of service is charged by the minute, then in this case each hash value represents a minute. When the MS is using this service provided by the SN, as each charging unit (in this case a minute) passes, the MS will sent one hash value, starting with R_{X-1}, to the SN in such order as $R_{X-1}, R_{X-2}, R_{X-2}, \cdots, R_1$. When the SN receives a new hash value, it hashes the value and sees if the result equals the hash value received last time. If so, the SN keeps each hash value for later charging because the hash values have to be presented to the ledger to prove the subscriber's access to the service. As Figure 5 shows, in the charging phase, the MS sends $E_{CK}(R_{X-i})$ (E is an encrypting function) as proof to the SN. During the message transmission between the MS and the SN, the cipher key CK in the UMTS AKA protocol is used to offer security guarantee. Please note all that the SN and the HN have to keep is the very last hash value received. When the service comes to an end or when a certain preset time interval is reached, the SN sends the last hash

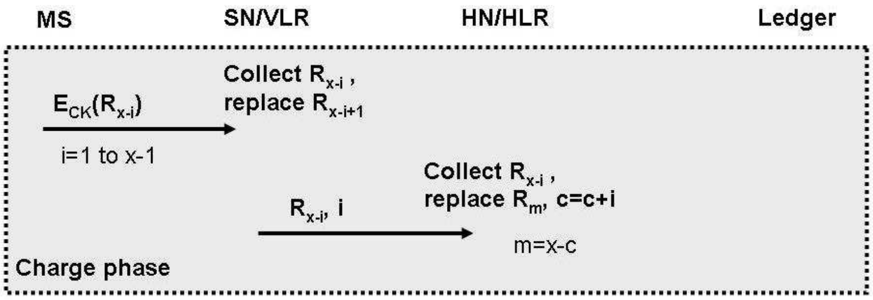

Figure 4: Our charging protocol

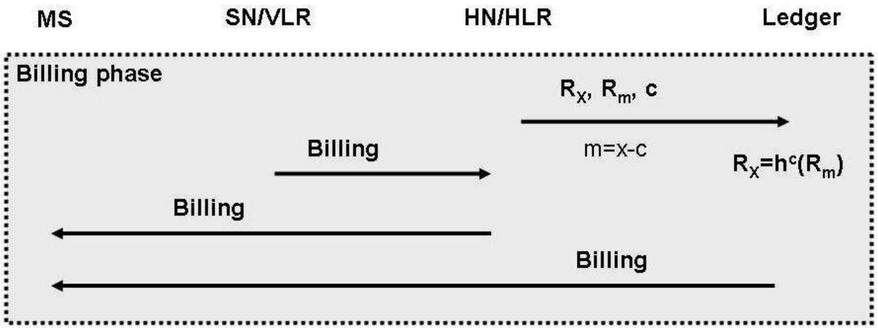

Figure 5: Our billing protocol

value R_{X-i} it received along with the count of charging units i to the HN. Taking down the records, the HN proves the correctness of the bill to the subscriber in the billing phase.

As Figure 5 shows, when it is time for the bill, the HN first checks to see if all the hash values match the subscriber's consumption records. If nothing goes wrong, the bill is sent to the MS, and the hash values are sent to the ledger server as proofs. The ledger server checks the correctness of both R_X and R_m and makes sure that the data came from the subscriber. Then, the hash values are used to figure out the amount of money to charge, and the subscriber is or course informed of the amount. On the other hand, if there are charging relationships between the SN and the HN, the SN will certainly send a bill to the HN too. The MS compares the bill received from the HN and the amount learned from the ledger server and sees if they show the same number. Everything proves to be all right if they are the same; however, if they differ, that is where our dispute protocol (see Figure 6) comes into play.

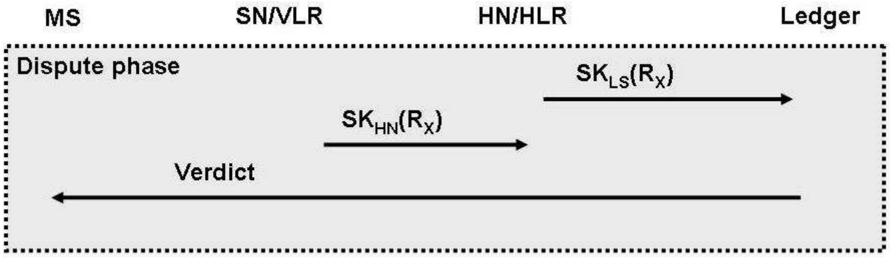

Figure 6: Our dispute protocol

The discrepancy that happens between the bill the MS receives from the HN and the message from the ledger server telling the MS how much he/she is supposed to pay comes most probably from the fact that the R_X actually came from somebody else but the MS. Therefore, to solve a dispute, all that the HN has to do is send back the $SK_{LS}(R_X)$ signed by the ledger server so that the ledger server can check out the R_X. By the same token, when a dispute occurs between the SN and the HN over the bill, the $SK_{HN}(R_X)$ signed by the HN can readily be sent back to the original signer to see if the R_X is the real thing. Due to the simple truth that R_0 is only known to the MS, the only party that can figure out $R_1 \sim R_{X-1}$ is the MS. As a result, if the HN can give the correct $SK_{LS}(R_X)$ and R_m that satisfy $h^C(R_m) = R_X$, there is no reason why the ledger server should doubt the genuineness of this proof.

3 Discussions and Analyzes

The currently used UMTS charging and billing system has been in service for many a year. What has been the standard way of doing it is to take down the records of the subscriber's accesses to the services to put together the CDR, which is then passed to the billing unit for the basic calculations to be done before the bill comes out to the subscriber. As we mentioned earlier, most of the research concern focuses on the development of either a secure subscriber authentication mechanism for fear of person or an integrated independent charging framework that avoids redundancies of the CDR. This way or that, however, the very basis is still the seemingly irreplaceable CDR. Our approach here to improving the UMTS charging and billing system is not to turn over the original thing, including the CDR foundation it relies on, but only to add a little something to it so that a long unsolved problem, or a long unaddressed issue, can be properly taken care of. The little extension we add to the original UMTS AKA protocol does accomplish its mission of offering the subscriber some solid grounds for fair play against the service provider on bill disputes, a right the subscriber deserves. By bringing in the ledger server, our AKA charging protocol makes it possible for the subscriber to disagree on the bill with easy, hard proof in the hand.

3.1 Our Objectives

The three objectives of our AKA charging protocol are to fairly resolve bill disputes, to be applicable to the current UMTS standard environment, and to fit the various ways of UMTS-based payment. Here are some discussions as to how the three objectives can be achieved.

1. Fair resolvability of bill disputes: Taking advantage of the non-repudiation property the hash chain has, we bring in hash values as proofs of service consumption. When there are disputes, these hash values are the best evidence the service provider has to prove the correctness of the bill to the subscriber. To rule out the possibility of the subscriber deceptively denying accessing the service, the ledger server will sign for the subscriber in the first place, which puts non-repudiation on the subscriber's consumption behaviors. In our protocol, non-repudiation does not have to show on the service provider's side against the subscriber; all the service provider has to do is make sure that the subscriber cannot deny any line on the bill. Anyways, without the right hash values, there is no way the service provider can charge the subscriber for any service that has not been used.

2. Applicability to the current UMTS standard environment: Our new protocol is only an extension added to the original UMTS AKA protocol with nothing

charged. The original system can function as it always does with or without our protocol taking effect. In other words, with our protocol, we are offering an option of provable charges and fair dispute resolvability added to all the original services.

3. Suitability for various ways of UMTS-based payment: As we mentioned earlier, in the UMTS design there are three ways of charging: Prepay Service, Real Time, and Session. In our AKA charging protocol, a subscriber can actually have more than one hash chain, each hash chain keeping track of a different way of charging at a certain charging rate.

    - Prepay Service: Suppose each hash value stands for a charging unit a, and the amount of money to be prepaid now is b. Given that $c = b/a$, and the first fresh hash value on the hash chain at this moment is R_m. The subscriber will send R_{m-c} to the service provider to prove that he/she is willing to prepay for the service and that the amount is b. Here, to make sure that the division b/c leaves no remainder, we can use different suitable hash chains when charging units differ.
    - Real Time: This is a more commonly used way of charging. When the service is in use, a hash value is sent out every fixed period of time, meaning that a charging unit of time has passed. The service provider only has to store the last hash value collected during each connection because it stands for the total time the connection lasts.
    - Session: This way of charging is very much similar to what is done for real time charging. The only difference is that with session charging, a hash value is transmitted along with every packet (or every fixed number of packets).

As for the charging relationships among the entities involved in the UMTS design is detal in [1], including interconnect charging, roaming charging, usage charging, conveyance charging, wholesale charging, supplier charging, and capacity charging, disputes in these places can all be avoided when our AKA charging protocol is applied.

3.2 Security

Before addressing the security issue, let's check out the whole background environment first. In the UMTS framework design, the MS translates to a portable, extremely light weight device with relatively limited computation and storage power. By contrast, the SN, HN, and the ledger server are fixed equipment with much greater computation and storage power. The communication between the MS and the SN depends on the radio wave, which is quite vulnerable to eavesdropping and interception launched by malicious attackers. Distinctly, the channel of communication built up to connect such fixed facilities as the SN, HN and ledger server is a wire through which the traffic runs in an encrypted form. This is a way of communication malicious attackers would consider more difficult to break in. Due to the limited computation and storage power the MS has, the ledger server is trusted with such tasks as signing and storing the proofs of service consumptions as well as checking the correctness of the bill. The following are some aspects of the security issue:

1. Authentication: Our AKA charging protocol is developed upon the basis of the original UMTS AKA protocol design. Therefore, as far as authentication is concerned, the security level of our AKA charging protocol is the same as that of the original AKA protocol, which is constructed by mechanisms such as mutual authentication, anonymity, and so on.

2. Interception: During transmission, the service consumption record R_{X-i} is always in its encrypted form, and the cipher key CK, which differs for each connection, is owned only by the MS and the SN. Therefore, there is no risk of information leaks due to data interception. As for the charging and billing

process, communication is done in the encrypted form through a wire within the facility, so no risk of information leaks is run, either.

3. Falsification: With our AKA charging protocol applied, there is no forging a fake R_{X-i}, and the reason for that is the non-repudiation property of the hash function. Although the SN and the HN have R_X, there is no way for them to forge a R_{X-i}; To obtain this R_{X-i}, the only possibility is to receive it from the MS as a record of service consumption, which of course does not come for no reason. As long as the service consumption record R_{X-i} received this time does not equal the hash value of the record R_{X-i+1} received last time, namely $R_{X-i} \neq (R_{X-i+1})$, the SN will refuse to provide any service because something has gone wrong.

4. Non-repudiation: The subscriber cannot deny sending out a service consumption record once he/she has. Before the service provider offers the service the MS asks for, it will receive a signature $SK_{LS}(R_X)$ signed by the ledger server, namely the proxy of the MS. Therefore, when it is time to pay the bill, the HN presents R_{X-i} to the MS as the proof of service consumption, and the MS cannot repudiate it as long as the equation $h^i(R_{X-i}) = R_X$ holds, because nobody but the MS is aware of R_0 and can thus figure out R_{X-i}.

3.3 Practicability

Practicability was a huge factor we took into account when we put together our AKA charging protocol. After all, UMTS is a technology of great convenience that has already been an indispensable part of modern life, and therefore, before we can actually make a difference and bring even greater convenience, the only thing that seems to matter is whether or not our new protocol can really fit in with the original UMTS. As we have been stressing, our new protocol is an extension from the UMTS AKA protocol and is thus perfectly compatible with the UMTS framework currently prevalently in use. As a matter of fact, the new protocol we present here can equip the service provider with the capability to offer subscribers a fairer, more convincing, more user-friendly charging and billing service.

Besides compatibility, the load on the system is also one thing to keep in mind. In order not to put too much of an extra load on the system, we have employed methods of low computation power consumption. Most of the computations our new protocol requires are hashing operations because the hash function is quite economical in computation power consumption. On the other hand, because what the MS has in the hand is a mobile device of low computation power, the ledger server, which is a fixed organization with great computation power, comes in and does the signing on behalf of the MS so that the non-repudiation requirement is satisfied. In fact, both the ledger server and the HN are capable of signing data when necessary.

In addition to the computational load, the storage space is also among the list of consideration. In spite of the fact that all of the SN, HN, and ledger server are fixed facilities seemingly free from storage space constraints, we still try to consume as little of it as possible. For the SN and the HN, the last hash value is the only one to store for each connection because all those that came before it can be derived from it. The focus of the concern, however, is on the MS. Due to the limited storage space the mobile device, in our protocol design, has to have the ledger server do the necessary data recordings. As far as the MS is concerned, there is still one more thing worthy of notice. We know that the MS owns the first hash value R_0 and can derive the whole hash chain R_X by repeatedly using the hash function following the order of $R_1, R_2, \cdots, R_{X-2}, R_{X-1}$. However, the hash values are in fact to be sent out in such order as $R_{X-1}, R_{X-2}, \cdots, R_2, R_1$. Generally speaking, the MS can either figure out all the hash values in the hash chain $(R_1, R_2, \cdots, R_{X-2}, R_{X-1})$ once and for all and store them for future use, or store only R_0 and compute R_{X-i} every time when it is time to use it. The former way saves time at the cost of plenty of storage space, while the latter way avoids the trouble of storage space insufficiency at the sacrifice of a lot more computation time. Fortunately, we have already proposed a solution to this tradeoff problem in our earlier research effort [30], striking a balance between computation efficacy and storage space economy.

Lastly, the Unbalanced One-Way Binary Tree [38] proposed by Yen et al. in 1999 will be a good scheme to put to use when the MS holds many hash chains at the same time as service consumption records representing various ways of payment and currencies.

4 Conclusion

Disputes occur every once in a while over the bill between the telecom company and the subscribers. Most of the times, the subscribers would grudgingly drop the case because they have trouble proving the mistakes on that bill. This does not seem fair at all, and nor does it make any sense. All the subscribers' resentment comes from the simple fact that the design of the charging and billing protocol is obviously in favor of the telecom company; the service consumption records and the right to charge calculations are both held in the telecom company's hand. As a consequence, the subscribers, smelling something fishy about the bill, can only argue barehanded, and that just does not bring them anywhere. To have the table turned for the subscribers, in this paper, we have presented a new AKA protocol that offers the subscribers their chance to hold their own proofs of service consumption. The new protocol is in fact an extension to the original UMTS AKA protocol that is perfectly compatible with the old system, a new way to improve the overall design that has never been tried before. Most importantly, with some additional parameters attached to the original, our new protocol is indeed capable of lifting the subscribers to an even ground against the telecom company.

In the design of out AKA charging protocol, we put in a new entity, namely the ledger server, whose role is a proxy of the subscriber worthy of his/her trust. Due to the low computation power and small storage space the subscriber's mobile device has, the ledger server has to sign and check the service consumption records on behalf of the subscriber. The existence of the ledger server in the new design makes our new protocol applicable to the current environment. Besides the subscriber and the telecom company, other related service providers involved such as roam service providers can also make use of our new protocol to ensure the correctness of their charges and bills. In a word, our extended AKA protocol is indeed a good solution to the charging and billing problems long left unsettled.

References

[1] 3rd Generation Partnership Project. "Technical specification group services and systems aspects, charging and billing,". tech. rep., 3GPP TS 22.115.

[2] 3rd Generation Partnership Project. "Technical specification group services and systems aspects, charging management, charging architecture and principles,". tech. rep., 3GPP TS 32.240.

[3] 3rd Generation Partnership Project. "Technical specification group services and systems aspects, charging management, charging principles,". tech. rep., 3GPP TS 32.200.

[4] 3rd Generation Partnership Project. "Technical specification group services and systems aspects, security architecture,". tech. rep., 3GPP TS 33.102.

[5] 3rd Generation Partnership Project. "Technical specification group services and systems aspects, specification of the MILENAGE algorithm set, document 1: General,". tech. rep., 3GPP TS 35.205.

[6] 3rd Generation Partnership Project. "Technical specification group services and systems aspects, specification of the MILENAGE algorithm set, document 5: Summary and results of design and evaluation,". tech. rep., 3GPP TS 35.909.

[7] 3rd Generation Partnership Project. “Technical specification group services and systems aspects, telecommunication management, charging management, 3G call and event data for the circuit switched (CS) domain,”. tech. rep., 3GPP TS 32.005.

[8] 3rd Generation Partnership Project. “Technical specification group services and systems aspects, telecommunication management, charging management, call and event data for the packet switched (PS) domain,”. tech. rep., 3GPP TS 32.015.

[9] 3rd Generation Partnership Project. “Technical specification group services and systems aspects, telecommunication management, charging management, charging data description for the circuit switched (CS) domain,”. tech. rep., 3GPP TS 32.205.

[10] 3rd Generation Partnership Project. “Technical specification group services and systems aspects, telecommunication management, charging management, charging data record (CDR) transfer,”. tech. rep., 3GPP TS 32.295.

[11] B. Askwith, M. Merabti, and Q. Shi, “MNPA: a mobile network privacy architecture,” *Computer Communications*, vol. 23, pp. 1777–1788, Dec. 2000.

[12] R. Bird, I. Gopal, A. Herzberg, P. Janson, S. Kutten, R. Molva, and M. Yung, “The kryptoknight family of light-weight protocols for authentication and key distribution,” *IEEE/ACM Transactions on Networking*, vol. 3, pp. 31–41, Feb. 1995.

[13] H.-B. Chen and S.-C. Hsueh, “Light-weight authentication and billing in mobile communications,” *IEEE 37th Annual 2003 International Carnahan Conference on Security Technology (Annual 2003)*, pp. 245–252, Oct. 2003.

[14] Yu-Yi Chen, Jinn-Ke Jan, and Chin-Ling Chen, “A fair and secure mobile billing system,” *Computer Networks*, vol. 48, pp. 517–524, July 2005.

[15] 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, pp. 99–111, July 2007.

[16] European Telecommunication Standards Institute (ETSI). "Recommendation GSM 03.20, security related network functions,". tech. rep., June 1993.

[17] F. Eyermann, P. Racz, B. Stiller, C. Schaefer, and T. Walter, "Service-oriented accounting configuration management based on diameter," *The 30th Anniversary IEEE Conference on Local Computer Networks*, pp. 621–623, Nov. 2005.

[18] M. G. Gouda1 and A. X. Liu, "Formal specification and verification of a secure micropayment protocol," *International Journal of Network Security*, vol. 7, pp. 81–87, July 2008.

[19] G. Horn, K.M. Martin, and C.J. Mitchell, "Authentication protocols for mobile network environment value-added services," *IEEE Transactions on Vehicular Technology*, vol. 51, pp. 383–392, Mar. 2002.

[20] M. S. Hwang and P. C. Sung, "A study of micro-payment based on one-way hash chain," *International Journal of Network Security*, vol. 24, no. 2, 2006, MONTH =.

[21] P. Janson, G. Tsudik, and M. Yung, "Scalability and flexibility in authentication services: the kryptoknight approach," *Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, pp. 725–736, Apr. 1997.

[22] L. Jin, S. Ren, L. Feng, and G.Z. Hua, "Research on WAP clients supports SET payment protocol," *IEEE Wireless Communications*, vol. 9, pp. 90–95, Feb. 2002.

[23] A. Karygiannis, A. Kiayias, and Y. Tsiounis, "A solution for wireless privacy and payments based on e-cash," *First International Conference on Security*

and Privacy for Emerging Areas in Communications Networks (SecureComm 2005), pp. 206–218, Sep. 2005.

[24] M. Koutsopoulou and A. Kaloxylos, “A holistic solution for charging, billing and accounting in 4g mobile systems,” *2004 IEEE 59th Vehicular Technology Conference (VTC 2004-Spring)*, vol. 4, pp. 2257 – 2260, May 2004.

[25] M. Koutsopoulou, A. Kaloxylos, A. Alonistioti, and L. Merakos, “A platform for charging, billing, and accounting in future mobile networks,” *Computer Communications*, vol. 30, pp. 516–526, Feb. 2007.

[26] Shiqun Li, Guilin Wang, Jianying Zhou, and Kefei Chen, “Undeniable mobile billing schemes,” in *4th European PKI Workshop: Theory and Practice (EuroPKI’07)*, vol. 4582, pp. 338–34, Mallorca, Balearic Islands, June 2007. LNCS, Springer-Verlag.

[27] Kumar Mangipudi, Rajendra Katti, and Huirong Fu2, “Authentication and key agreement protocols preserving anonymity,” *International Journal of Network Security*, vol. 3, pp. 259–270, Nov. 2006.

[28] S. Mohanty and J. Xie, “Performance analysis of a novel architecture to integrate heterogeneous wireless systems,” *Computer Networks*, vol. 51, pp. 1095–1105, Mar. 2007.

[29] B.C. Neuman, “Security, payment, and privacy for network commerce,” *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 1523–1531, Oct. 1995.

[30] H.-H. Ou, M.-S. Hwang, and J.-K. Jan, “A simple mobile communication billing system among charged parties,” *Applied Mathematics and Computation*, vol. 192, Sep. 2007.

[31] D. Palaka, P. Daras, K. Petridis, and M. G. Strintzis, "A novel peer-to-peer payment protocol," *International Journal of Network Security*, vol. 4, pp. 107–120, Jan. 2007.

[32] M. Pias, S. Wilbur, S. Bhatti, and J. Crowcroft, "Securing the internet metering and billing," *IEEE Global Telecommunications Conference (GLOBECOM '02)*, vol. 2, pp. 1603–1607, Nov. 2002.

[33] J.F. Stach, E.K. Park, and K. Makki, "Performance of an enhanced GSM protocol supporting non-repudiation of service," *Computer Communications*, vol. 22, pp. 615–680, May 1999.

[34] H. Tewari and D. O'Mahony, "Real-time payments for mobile IP," *IEEE Communications Magazine*, vol. 41, pp. 126–136, Feb. 2003.

[35] H. Wang, Y. Zhang, J. Cao, and V. Varadharajan, "Achieving secure and flexible m-services through tickets," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 33, pp. 697–708, Nov. 2003.

[36] Shengbao Wang, Zhenfu Cao, and Haiyong Bao, "Efficient certificateless authentication and key agreement (CL-AK) for grid computing," *International Journal of Network Security*, vol. 7, pp. 342–347, Nov. 2008.

[37] Chou-Chen Yang, Kuan-Hao Chu, and Ya-Wen Yang, "3G and WLAN interworking security: Current status and key issues," *International Journal of Network Security*, vol. 2, pp. 1–13, Jan. 2006.

[38] S. Yen, L. Ho, and C. Huang, "Internet micropayment bared on unbalanced one-way binary tree," *Proceedings on CrypTEC '99*, pp. 155–162, 1999.

[39] S. Yu, S. Yoon, J. Lee, H. Kim, and J. Song, "Service-oriented issues: Mobility, security, charging and billing management in mobile next generation networks," *The 1st International Workshop on Broadband Convergence Networks (BcN 2006)*, pp. 1–10, Apr. 2006.