The Journal of Systems and Software 83 (2010) 316–325

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

A cocktail protocol with the Authentication and Key Agreement on the UMTS ☆

Hsia-Hung Ou [b], Min-Shiang Hwang [a,*], Jinn-Ke Jan [b]

[a] Department of Management Information Systems, National Chung Hsing University, 250, Kuo Kuong Road, Taichung 402, Taiwan, ROC
[b] Department of Computer Science and Engineering, National Chung Hsing University, Taichung 402, Taiwan, ROC

ARTICLE INFO

Article history:
Received 20 February 2008
Received in revised form 27 May 2009
Accepted 8 August 2009
Available online 21 August 2009

Keywords:
AKA
Authentication
Key agreement
UMTS

ABSTRACT

At present, the Universal Mobile Telecommunications System (UMTS) is very popular in most parts of the world. It is a third-generation mobile communication technique known for its ability to conduct user authentication and for its security of communication with the use of Authentication and Key Agreement (AKA) protocol. A mobile station (MS), a service network (SN) and a home environment (HE) use the protocol to authenticate each other and make an agreement with a session key. With the UMTS-AKA protocol standard, all authentication vectors (AV) produced by the HE are transferred to the SN for mutual authentication with the MS. In this scenario, authentication is exposed to two kinds of defects. One defect is computational overhead concentrating on the HE and the other is the communication overhead for delivering the AVs. To overcome these congenital defects, this study proposes a unique UMTS-AKA protocol called the cocktail-AKA protocol. The goal of this protocol is to allow the SN to share some medicated authentication vectors (MAV) that are calculated in advance and combined with a prescription at the authentication stage. So, the HE only needs to produce a prescription authentication vector (PAV). Once the authentication stage is initiated, the SN distributes MAV and PAV and produces an effective AV for mutual authentication with the MS. The cocktail-AKA protocol can overcome both the aforesaid defects.

1. Introduction

Cell phones are a significant innovations of the present-day world. The ease of carriage has made cell phones an essential part of the personal outfit in modern lifestyle. Currently, the Global System for Mobile communication (GSM) (ETSI, 1993) and the Universal Mobile Telecommunications System (UMTS) (3GPP, 2009) are the technologies most widely used world over. Presently, a third-generation (3G) cell phone technology, based on UMTS, which originated from GSM, is becoming popular to gradually replace GSM. The difference between cell phones and telephones is in their convenience. Cellular phones allow constant communication from virtually anyplace covered by the network of a base station (BS). A BS or service network (SN) is a stationary facility that receives and sends them telephonic signals between users. It acts as a medium between cellular phones and the home SN. Each BS covers an exclusive service region with a specific range of signals. However, it is possible for a mobile subscriber to move out of one's home BS and enter into the service region of another BS. In such circumstances, communication is possible by two ways: the subscriber has moved into the service region of the same SN, or the subscriber has moved into the service region of a third-party SN who is in partnership with the home SN. When a subscriber moves from one SN to another, he must prove his identity to both service providers and establish himself as a privileged user.

The UMTS's Authentication and Key Agreement (AKA) protocol (3GPP TS 33.102, 2006) was proposed by the 3GPP (3rd Generation Partnership Project) (3GPP, 2009) for this purpose. The authentication foundation of the AKA protocol comes into action when the subscriber's mobile station (MS) and the home environment (HE) enter into a secure key (K) agreement. In a practical situation, the MS is usually unable to establish HE authentication directly and must resort to another SN. The MS moves the authentication call with the SN, which ultimately obtains HE authentication. For this to happen, the SN must belong to the same network as the MS or to other networks with a partnership. This means that the SN and the HE must have prior security association. In the UMTS-AKA protocol, the SN should obtain authentication vectors (AV) from the HE of the MS when the MS moves into the service region of the SN and requests service. The HE produces all AVs and relinquishes them to the SN for mutual authentication with the MS. In the step, the HE calculates n-sets of AVs and then relinquishes the results to the SN. The SN uses this data to authenticate n-times with the MS. Generally, the SN uses one set of AV a time to facilitate the authentication of the MS. The following challenge/response method is being adopted in the UMTS-AKA protocol. First,

☆ 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. Tel.: +886 4 3323000; fax: +886 4 3742337.
E-mail address: mshwang@nchu.edu.tw (M.-S. Hwang).

0164-1212/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2009.08.019

the SN sends some data as a challenge to the MS, which than verifies the correctness of the data and calculates a reply response. Next, the SN compares the response of the MS with the parameters in the AV to verify the identity of the MS. Lastly, the MS calculates the relational encryption key from the challenge, and the SN deciphers the key from the AV. In this scenario, authentication is exposed to two kinds of congenital defects. One is computational overhead concentrating on the HE. The other is communication overhead for delivering the AVs.

This study proposes an innovative UMTS-AKA protocol, called the cocktail protocol, to solve the aforesaid defects. In this protocol, the SN shares some AVs (as the medicated authentication vectors, MAV) calculated in advance and combines them with the instructions at the authentication stage. The HE simply needs to produce an instruction authentication vector (as the prescription authentication vectors, PAV). When the authentication stage is initiated, the SN dispenses these two AV's and produces an effective authentication vector for mutual authentication with the MS. The cocktail protocol can reduce both computational overhead concentrating on the HE and communication overhead for delivering the AVs.

The remainder of this article is organized as follows: Section 2 introduces the concepts of UMTS-AKA protocol, reviews the literature, and points out the scope for its improvements. Section 3 presents the cocktail protocol as an improvement over the UMTS-AKA protocol. Section 4 provides relevant discussion and analysis. Finally, Section 5 presents the conclusion.

2. Related works

Many of research efforts have been directed to investigate the UMTS-AKA protocol, with a keen focus on security enhancements. A great deal of improvements has been studied to the cryptographic technology (Chaturvedi and Lal, 2008; Dimitriadis and Shaikh, 2007; Juang and Wu, 2008; Lee and Yeh, 2008; Mangipudi et al., 2006; Wang et al., 2008; Wang et al., 2008; Yang et al., 2006). These documents used symmetric cryptography, asymmetric cryptography or other methods to improve security. Given the outstanding characteristics of cryptography, it can be employed to reinforce the security of the UMTS-AKA protocol; however, the main disadvantage with it is higher computation overhead. Generally, the efficiency of mobile phones is limited by the complexity of computation. Although some high-priced phones can fulfill this demand, one should assimilate the fact that excessive equipment requirements can limit the popularity of the UMTS technology.

In 2003, Harn and Hsin used the concept of hash chain and MAC (message authentication code) to design an ER-AKA protocol (Harn and Hsin, 2003), which is expected to enhance the security of the original UMTS-AKA protocol. However, the protocol has greatly increased space overhead and communication overhead in the storage and transmission of hash chain. In 2005, Zhang and Fang cited some failures in the UMTS-AKA protocol, such as redirection attack and active attack in corrupted networks. In view of this, AP-AKA (Zhang and Fang, 2005) was introduced to meet higher security requirements. The newer protocol offered a solution of combining the service network's identity within the authentication token to prevent redirection attack and active attack in corrupted networks. However, it adopts a way similar as the original UMTS-AKA protocol to inherit the original shortcoming, that is, computation overhead and communication overhead in calculating and transmitting AVs. In 2005, Huang and Li introduced the X-AKA protocol (Huang and Li, 2005) to overcome the program of low bandwidth consumption. In this protocol, the MS's HE distributes a TK (temporary key) to SN. The SN uses a TK to carry out mutual authentication between the SN and the MS. This is different from the original method in that the HE calculates n-sets of AV and passes them to the SN. A TK is much smaller than n-sets of AV, and therefore it can save bandwidth consumption. In fact, the approach of this protocol is to transfer the HE's computation overhead to the SN, so as to reduce communication overhead. However, a closer look at the design of UMTS-AKA protocol reveals that the SN almost makes no computation expect that it compares the messages. It can speculate that the SN may not be able to afford complex computations with the current equipment. Therefore, this proposal may not be suitable for the existing environment. By virtue, the designs of AP-AKA and X-AKA were expected to adhere to the UMTS standard. Unfortunately, both use a number of custom encryption functions. These may interfere with the practicability in these protocols combining with the current mechanism. Recently, in 2006, AI-Saraireh and Yousef suggested some remedies to avoid the bottlenecks of the UMTS-AKA protocol, which include reducing the number of messages transmitted between mobile phones and authentication center and minimizing authentication time delay, call setup time and signaling traffic. Their proposal Al-Saraireh and Yousef (2006) has reversed the original authentication process, that is, AVs are not generated by the HE but instead the MS sends the authentication token via SN to the HE. In accordance with the authors' simulation, signaling messages between the mobile network entities are reduced and moreover, authentication time delay, call setup time and signaling traffic are minimized as compared to the original UMTS-AKA protocol. However, their protocol showed two deadly mistakes. One is that the MS must save n-sets of AVs on a limited space in mobile devices. The other mistake is that each authentication SN must pass the authentication token back to the HE; this will cause delay in authentication.

It is not the aim of this study to design the safest or the best AKA protocol. Indeed, the purpose is to design a most practical AKA protocol. It is very important to adopt the same technology as the current UMTS-AKA protocol. It ensures painless transfer to a new environment without extra cost. In addition, the attractive benefits of other literatures with are also been proposed on this study.

3. Review of the UMTS-AKA protocol

Fig. 1 illustrates UMTS-AKA protocol, and Fig. 2 lists the abbreviations and notations used in the UMTS-AKA protocol. The UMTS-AKA protocol has two phases. Phase 1 involves the distribution of AVs, and Phase 2 involves authentication and key establishment. In phase 1, the SN receives AVs by delivery of the MS's IMSI (International Mobile Subscriber Identity) to the HE. In Phase 2, there are n-sets of AV's for n-times of AKA between the MS and the SN.

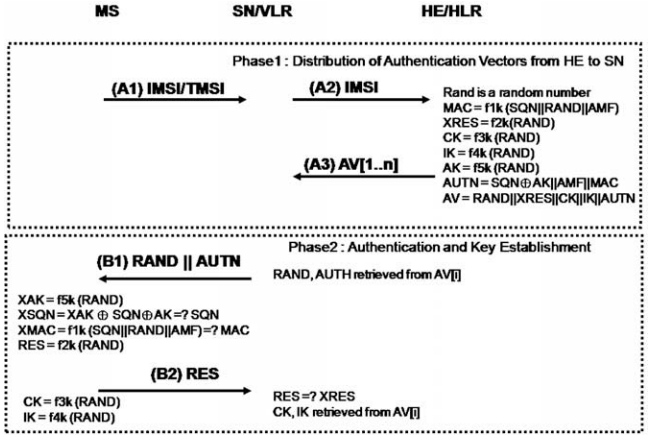

Fig. 1. The UMTS-AKA protocol.