Technical University Berlin

Telecommunication Networks Group

Current Approaches to Authentication in Wireless and Mobile Communications Networks

G. Schäfer, A. Festag, H. Karl
[schaefer,festag,karl]@ee.tu-berlin.de

Berlin, 26/03/2001, Version 1.0

TKN Technical Report TKN-01-002

TKN Technical Reports Series

Editor: Prof. Dr.-Ing. Adam Wolisz

Abstract

This document[1] gives a brief introduction into algorithms and protocols for *entity authentication* (verifying the identity of communication partners) and analyzes the approaches for realizing authentication in current mobile communication standards. The main results of this comparative analysis concerning an authentication infrastructure for wireless Internet access are, that (1) the protocols as proposed in current IETF working groups still need further evaluation of their security characteristics, and, in particular, (2) do exhibit serious deficiencies regarding the location privacy of mobile nodes. Furthermore, it is concluded that in order to assess the performance implications of (re-)authentication during frequent handovers further study is needed which will be addressed in a future report.

[1]This work has been supported by a grant from Siemens AG.

Contents

Chapter 1

Introduction

Upcoming next generation wireless networks aim to support data and multimedia services to *mobile nodes (MNs)*, in principle regardless of the origin of the MN. “In principle” means that a visited network, e.g. a *radio access network (RAN)*, may require some kind of general or specific service agreement between its administrative domain and the MNs original administrative domain. This objective poses considerable requirements on the *authentication service* which assures that entities have in fact the *identity* they claim to have.

This report gives a brief introduction to authentication protocols and analyzes the approaches to authentication of current mobile communication standards including ongoing work of the *Internet Engineering Taskforce (IETF)* regarding authentication for *Mobile IP*. Basic familiarity with the concepts of Mobile IP is assumed (see e.g. [9] for a brief introduction).

The remaining sections of this chapter introduce basic terminology and address some initial architectural considerations for future IP based RANs as well as different usage scenarios for Mobile IP in this context. Chapter 2 gives a brief survey over basic authentication primitives. Chapter 3 analyzes the authentication protocols of current mobile communication standards, and Chapter 4 draws some conclusions concerning our future work in this area.

1.1 Basic Terminology

The following definitions are taken from two internet drafts that address the requirements of authentication, authorization, and accounting (AAA) as currently discussed in the IETF [2, 11].

- *Accounting:* The act of collecting information on resource usage for the purpose of trend analysis, auditing, billing or cost allocation.
- *Administrative Domain:* An intranet, or a collection of networks, computers and databases under a common administration. Computer entities operating in a common administration may be assumed to share administratively created security associations.
- *Attendant:* A node designed to provide the service interface between a client and the local

domain, e.g. a Mobile IP *foreign agent (FA)* or a *network access server (NAS)* offering *point-to-point tunneling protocol (PPP)* service.

- *Authentication:* The act of verifying a claimed identity, in the form of a pre-existing label from a mutually known name space, as the originator of a message (message authentication) or as the end-point of a channel (entity authentication).
- *Authorization:* The act of determining if a particular right, such as access to some resource, can be granted to the presenter of a particular credential.
- *Billing:* The act of preparing an invoice.
- *Broker:* An intermediary agent, trusted by two other AAA servers, able to obtain and provide security services from those AAA servers. For instance, a broker may obtain and provide authorizations, or assurances that credentials are valid.
- *Client:* A node wishing to obtain service from an attendant within an administrative domain.
- *End-to-End:* End-to-End is the security model that requires that security information be able to traverse, and be validated even when an AAA message is processed by intermediate nodes such as proxies, brokers, etc.
- *Foreign Domain:* An administrative domain, visited by a Mobile IP client, and containing the AAA infrastructure needed to carry out the necessary operations enabling Mobile IP registrations. From the point of view of the foreign agent, the foreign domain is the local domain.
- *Home Domain:* An administrative domain, containing the network whose prefix matches that of a mobile node's home address, and containing the AAA infrastructure needed to carry out the necessary operations enabling Mobile IP registrations. From the point of view of the home agent, the home domain is the local domain.
- *Hop-by-hop:* Hop-by-hop is the security model that requires that each direct set of peers in a proxy network share a security association, and the security information does not traverse an AAA entity.
- *Inter-domain Accounting:* Inter-domain accounting is the collection of information on resource usage of an entity with an administrative domain, for use within another administrative domain. In inter-domain accounting, accounting packets and session records will typically cross administrative boundaries.
- *Intra-domain Accounting:* Intra-domain accounting is the collection of information on resource usage within an administrative domain, for use within that domain. In intra-domain accounting, accounting packets and session records typically do not cross administrative boundaries.
- *Local Domain:* An administrative domain containing the AAA infrastructure of immediate interest to a Mobile IP client when it is away from home.
- *Proxy:* An AAA proxy is an entity that acts as both a client and a server. When a request is received from a client, the proxy acts as an AAA server. When the same request needs to be forwarded to another AAA entity, the proxy acts as an AAA client.

- *Local Proxy:* A Local Proxy is an AAA server that satisfies the definition of a Proxy, and exists within the same administrative domain as the network device (e.g. NAS) that issued the AAA request. Typically, a local proxy will enforce local policies prior to forwarding responses to the network devices, and are generally used to multiplex AAA messages from a large number of network devices.

- *Network Access Identifier:* The Network Access Identifier (NAI) is the userID submitted by the client during network access authentication. In roaming, the purpose of the NAI is to identify the user as well as to assist in the routing of the authentication request. The NAI may not necessarily be the same as the user's e-mail address or the user-ID submitted in an application layer authentication.

- *Routing Broker:* A Routing Broker is an AAA entity that satisfies the definition of a Broker, but is NOT in the transmission path of AAA messages between the local ISP and the home domain's AAA servers. When a request is received by a Routing Broker, information is returned to the AAA requester that includes the information necessary for it to be able to contact the Home AAA server directly. Certain organizations providing Routing Broker services MAY also act as a Certificate Authority, allowing the Routing Broker to return the certificates necessary for the local ISP and the home AAA servers to communicate securely.

- *Proxy Broker:* A Proxy Broker is an AAA entity that satisfies the definition of a Broker, and acts as a Transparent Proxy by acting as the forwarding agent for all AAA messages between the local ISP and the home domain's AAA servers.

- *Roaming Capability:* Roaming capability can be loosely defined as the ability to use any one of multiple Internet service providers (ISPs), while maintaining a formal, customer-vendor relationship with only one. Examples of cases where roaming capability might be required include ISP "confederations" and ISP-provided corporate network access support.

1.2 Architecture Assumptions

Current discussions about next-generation mobile networks favour an all-IP architecture for future RANs. It has to be stated that the partition of the RANs into IP subnets as well as the authentication policy and -mechanisms concerning re-authentication when changing IP subnets will have a particular impact on the performance of the authentication service. Therefore, this section lists some basic assumptions about the network architecture on which we will base our further work concerning authentication for Mobile IP[1]:

- We will assume a cellular network in which different cells are addressed in different IP subnetworks. This implies, for example, that a handover from one *base station controller (BSC)* to another one will result in a handover-operation in the IP layer and the *mobile node (MN)* will change its temporary care-of-address.

[1] This list will have to be extended in the course of the project.

- An eventually existing hierarchy of cells (macro- / micro- / pico-cells), e.g. motivated by the underlying link-layer technology, is not reflected in the IP addressing scheme in order to avoid fragmentation of IP address space. Consequently, there will probably be no implicit means to derive the number of the target IP subnetwork and a related security association in case of a vertical handover.
- A cellular network is equipped with some kind of *AAA-server* (e.g. a RADIUS, DIAMETER, or COPS server), which provides basic authentication, authorization and accounting services to the attendants of that cellular network.

Furthermore, the following Mobile IP deployment scenarios will have to be taken into account when evaluating an authentication and key management architecture for future All-IP based RANs.

- The home network of a mobile node may be a private IP subnet of a company, or a public subnet of a service provider.
- It might be desirable to allocate the home agent in the visited network if the MN does not require to get an IP address of “his home network”.

Chapter 2

Principles of Authentication

Authentication, the proof of the identity of an entity or the origin of a message, is the most fundamental security service as all other security services build upon it. The following two principle variants of authentication have to be distinguished:

1. *Data origin authentication* is the security service that enables entities to verify that a message has been originated by a particular entity and that it has not been altered afterwards. A synonym for this service is *Data Integrity*. The relation of data integrity to cryptographic protocols is twofold:
   - There are cryptographic protocols to ensure data integrity. As a rule, they comprise just one protocol step and are, therefore, not very "exciting" from a protocol point of view.
   - Data integrity of messages exchanged is often an important property in cryptographic protocols, so data integrity is a building block to cryptographic protocols.
2. *Entity authentication* is the security service, that enables communication partners to verify the identity of their peer entities. In principle, it can be accomplished by various means:
   - Knowledge: e.g. passwords
   - Possession: e.g. physical keys or cards
   - Immutable characteristic: e.g. biometric properties like fingerprint, etc.
   - Location: evidence is presented that an entity is at a specific place (example: people check rarely the authenticity of agents in a bank)
   - Delegation of authenticity: the verifying entity accepts, that somebody who is trusted has already established authentication

   As in communication networks, direct verification of the above means is difficult or insecure, cryptographic protocols have been developed for this purpose.

This chapter will introduce some general background on cryptographic algorithms and on authentication protocols, that use these algorithms as basic building blocks. As the security properties of authentication protocols are more difficult to assess than other properties of communication protocols,

the formal evaluation of authentication protocols is treated as well. The main purpose of this chapter is to give sufficient background for the discussion of current approaches to authentication in mobile networks in the remainder of this report and the further course of the project.

2.1 Cryptographic Algorithms

There are two main categories of cryptographic algorithms which serve as fundamental building blocks of authentication protocols:

- *Encryption Algorithms*, which may be further divided into:
  - *Symmetric Encryption Algorithms*, that use a single key for encryption and decryption of data. This key has to be kept *secret* between two entities participating in a secure exchange, explaining the common term *secret key encryption* for this class of algorithms. Prominent algorithms in this category are the *Data Encryption Standard (DES)* [33] and the *International Data Encryption Algorithm (IDEA)* [22].
  - *Asymmetric Encryption Algorithms*, that use two different keys for encryption and decryption of data. Each entity possesses a pair of keys: one *private key* which is only known to itself and one *public key* which is publicly announced, explaining the commonly used term *public key cryptography* for this class of algorithms. If a sending entity wants to make sure, that a message can just be read by the intended receiver of the message, it encrypts the message with the public key of that receiver. As the corresponding private key is only known to the receiver, he is the only one being able to decrypt the message. Furthermore, the sender can as well encrypt the message with his private key. This does not protect the secrecy of the message, as the corresponding public key is in principle known to everyone, but allows every other entity to verify, that the message has in fact been originated by the sender, as only he knows his private key. Prominent examples for asymmetric encryption algorithms are the *Rivest-Shamir-Adleman (RSA)* algorithm [27, section 8.2] and the *ElGamal* algorithm [27, section 8.4].
- *Integrity Check Values*, that may be further subdivided into:
  - *Modification Detection Codes (MDC)* which are computed using a class of algorithms called *Cryptographic Hash Functions*. An MDC alone does not protect a message, it represents a digital fingerprint which needs to be signed using either a secret or a private key in order to ensure, that the message originated from a specific sender. Common algorithms in this class are the *Message Digest 5 (MD5)* and the *Secure Hash Algorithm (SHA-1)* [42, chapter 9].
  - *Message Authentication Codes (MAC)* which are computed over a message making use of a secret key and, therefore, directly allow to ensure that a message has been originated by one of the entities knowing that key. A very common algorithm for computing a MAC is to use a symmetric block cipher like DES or IDEA in a special mode called *Cipher Block Chaining Mode* and to use the output block of the encryption process as the MAC. An alternative but controversially discussed method is to “mix” some secret value with the

Table 2.1: Notation of Cryptographic Protocols

Notation	Meaning
A	Name of A, analogous for B, TTP and CA
CA_A	Certification Authority of A
r_A	Random value chosen by A
t_A	Time-stamp created by A
$(m_1, \ldots, m_n)$	Concatenation of messages m_1 to m_n
$A \rightarrow B : m$	A sends message m to B
$K_{A,B}$	Symmetric key shared by A and B
$+K_A$	Public Key of A
$-K_A$	Private Key of A
$\{m\}_K$	Message m encrypted with key K
$H(m)$	Hash value of message m
$A[m]$	Shorthand notation for $\left(m, \{H(m)\}_{-K_A}\right)$
$Certificate_{-CK_{CA}}(+K_A)$	Certificate of $+K_A$ issued by CA
$CA << A >>$	Shorthand notation for $Certificate_{-CK_{CA}}(+K_A)$

message and to compute an MDC over the resulting message. Somebody, who does not know the secret value is supposed not to be able to calculate a matching MDC, but there is some cryptographic concern over this (cf. [27, section 9.5.2]). A construction that is (up to now) considered to be secure is HMAC [21] (see also [27, note 9.67]).

2.2 Entity Authentication Protocols

Entity authentication protocols can be subdivided into two main categories:

- *Arbitrated Authentication*, in which two (or more) entities, that want to verify the authenticity of one or more entities make use of a so-called *trusted third party (TTP)*, and
- *Direct Authentication*, in which the authentication is handled without direct involvement of a trusted third party.

Please note, that also in direct authentication there might exist a trusted third party, e.g. certifying the authenticity of public keys. The difference between both categories is, that in arbitrated authentication the trusted third party needs to participate in every authentication exchange, whereas direct authentication does not require the online presence of the TTP.

In order to allow for a unified presentation of authentication protocols table 2.1 introduces some notation which will be used in the remainder of this report.

2.2.1 Arbitrated Authentication

In order to illustrate the basic ideas of arbitrated authentication, we give two prominent examples of protocols of this class, the *Needham-Schroeder Protocol* and the authentication and access control system *Kerberos*.

The Needham-Schroeder Protocol

The Needham-Schroeder protocol [32] allows two entities Alice (A) and Bob (B) to authenticate each other by the help of a trusted third party (TTP). The protocol uses a symmetric encryption algorithm as basic cryptographic primitive. The trusted third party holds a database of all users U which want to make use of its authentication service and this database also contains a secret key $K_{U,TTP}$ for each user U. The goal of a protocol run is to authenticate two users A and B and to establish a secret key for securing further communication between them. The protocol proceeds as follows:

1. Alice chooses a random number r_A, creates a message containing her name A, Bob's name B and the random number, and sends this message to TTP:

$$A \rightarrow TTP: \quad \left(A, B, r_A\right) \tag{2.1}$$

2. TTP generates a session key $K_{A,B}$ for secure communication between A and B, encrypts this key together with the name of A using the key $K_{B,TTP}$ it shares with B, and sends the following message encrypted with the key $K_{A,TTP}$ to A:

$$TTP \rightarrow A: \quad \left\{r_A, B, K_{A,B}, \{K_{A,B}, A\}_{K_{B,TTP}}\right\}_{K_{A,TTP}} \tag{2.2}$$

3. Alice decrypts this message, checks that the random number r_A is the same as in her first message and sends the following message to Bob:

$$A \rightarrow B: \quad \left\{K_{A,B}, A\right\}_{K_{B,TTP}} \tag{2.3}$$

4. Upon reception of this message Bob decrypts it, generates a random number r_B, encrypts this number with $K_{A,B}$ and sends it to Alice:

$$B \rightarrow A: \quad \left\{r_B\right\}_{K_{A,B}} \tag{2.4}$$

5. Alice decrypts the message with $K_{A,B}$, computes $r_B - 1$, encrypts the result with $K_{A,B}$ and sends the result back to Bob:

$$A \rightarrow B: \quad \left\{r_B - 1\right\}_{K_{A,B}} \tag{2.5}$$

6. Upon reception, Bob decrypts the message, checks if it contains $r_B - 1$ and if so, assumes that Alice is authentic.

The two last messages serve the purpose, that Alice proves to Bob, that she in fact knows the key $K_{A,B}$. If not, she would not be able to compute $\{r_B - 1\}_{K_{A,B}}$. As Bob knows, that TTP sends his message, containing this key encrypted with the key $K_{A,TTP}$, he concludes that Alice knows the key $K_{A,TTP}$ and is therefore authentic.

However, this reasoning includes a flaw, which can be exploited by an attacker Eve (E) who once gets to know a valid session key $K_{A,B}$ [8]. As there is no means in the protocol to detect old session keys, that have already been used in a prior session, the attacker can authenticate himself to Bob as Alice by re-using an old session key, she somehow managed to figure out:

1. Eve sends the recorded message:
$$E \rightarrow B: \quad \left\{K_{A,B}, A\right\}_{K_{B,TTP}} \tag{2.6}$$
2. Upon reception of this message Bob decrypts it, generates a random number r_B, encrypts this number with $K_{A,B}$ and sends it to Alice:
$$B \rightarrow A: \quad \left\{r_B\right\}_{K_{A,B}} \tag{2.7}$$
3. Eve intercepts this message, decrypts it with $K_{A,B}$ (please note, that Eve needs to know $K_{A,B}$), computes $r_B - 1$, encrypts the result with $K_{A,B}$ and sends the result back to Bob:
$$E \rightarrow B: \quad \left\{r_B - 1\right\}_{K_{A,B}} \tag{2.8}$$
4. Upon reception, Bob decrypts the message, checks if it contains $r_B - 1$ and if so, assumes that Alice is authentic, i.e. Eve is Alice.

Please note, that the original intention of the protocol design was to ensure, that only entities possessing $K_{A,TTP}$ are able to authenticate as Alice. Because of the protocol flaw described above, it is sufficient to figure out one session key $K_{A,B}$ to authenticate to user B as Alice. As the key $K_{A,B}$ is potentially used to encrypt large amounts of data during the session following an authentication exchange, it might be easier to crypt-analyze this key, that the key $K_{A,TTP}$. Concluding, the original Needham-Schroeder protocol attains weaker security than originally intended.

It has, therefore, been revised by a couple of cryptographers including Needham and Schroeder themselves. Their solution [31] is essentially the same as the Otway-Rees protocol [34], published in the same journal:

1. Alice generates a message containing an index number i_A, her name A, Bob's name B, and the same information plus an additional random number r_A, encrypted with the key $K_{A,TTP}$ she shares with TTP and sends this message to Bob:
$$A \rightarrow B: \quad \left(i_A, A, B, \{r_A, i_A, A, B\}_{K_{A,TTP}}\right) \tag{2.9}$$
2. Bob generates a random number r_B, encrypts it together with i_A, A and B using the key $K_{B,TTP}$ he shares with TTP and sends the following message to TTP:
$$B \rightarrow TTP: \quad \left(i_A, A, B, \{r_A, i_A, A, B\}_{K_{A,TTP}}, \{r_B, i_A, A, B\}_{K_{B,TTP}}\right) \tag{2.10}$$

3. Upon reception, TTP decrypts the two encrypted sub-messages, generates a new session key $K_{A,B}$ and creates two encrypted messages, one for Alice and one for Bob, and sends them both to Bob:

$$TTP \rightarrow B: \quad \left(i_A, \{r_A, K_{A,B}\}_{K_{A,TTP}}, \{r_B, K_{A,B}\}_{K_{B,TTP}}\right) \tag{2.11}$$

4. After receiving this message, Bob decrypts his part of the message using $K_{B,TTP}$, verifies that r_B is identical to the random number he generated in the second step of the protocol and sends Alice's part of the message to her:

$$B \rightarrow A: \quad \left(i_A, \{r_A, K_{A,B}\}_{K_{A,TTP}}\right) \tag{2.12}$$

5. Upon reception of this message, Alice decrypts it with $K_{A,TTP}$ and verifies, if the contained random number r_A matches the one generated in the first step of the protocol. If she now uses the session key $K_{A,B}$ in an encrypted communication with Bob, she can be sure of his authenticity, as only TTP could have generated $\{r_A, K_{A,B}\}_{K_{A,TTP}}$ and and an eventual attacker Eve is not able to alter $\{r_A, i_A, A, B\}_{K_{A,TTP}}$ she generates in the first protocol step.

The same argument applies to Bob, so that he can conclude to communicate with Alice if he receives intelligible messages from her that have been encrypted with the session key $K_{A,B}$. However, it is important to note that both Alice and Bob need to completely trust in the correct functioning and honesty of TTP.

The Kerberos Authentication System

The Kerberos authentication system has been designed in the late 1980's in the course of the project *Athena* at the Massachusetts Institute of Technology (MIT), Boston, USA. Kerberos provides an authentication and access control service for workstation clusters. Its main design goals were the following:

- *Security:* eavesdroppers or active attackers should not be able to obtain the necessary information to impersonate a user when accessing a service.
- *Reliability:* as every use of a service requires prior authentication, Kerberos should be highly reliable and available.
- *Transparency:* the authentication process should be transparent to the user beyond the requirement to enter a password.
- *Scalability:* the system should be able to support a large number of clients and servers.

The basic usage scenario of Kerberos is a user, Alice, who wants to access one or more different services, that are provided by different servers $S1$, $S2$, ... connected over an insecure network. Kerberos deals with the following security aspects of this scenario:

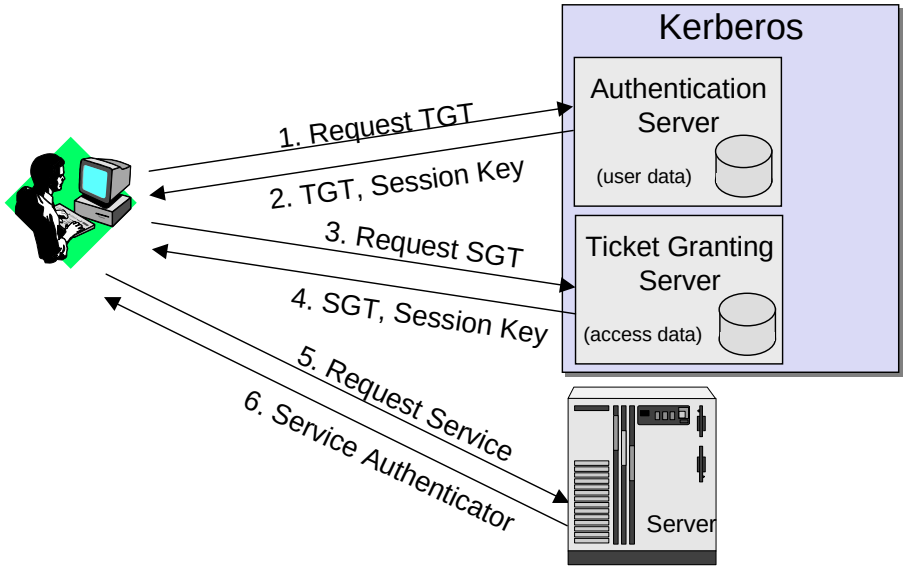


Figure 2.1: Overview over the Kerberos Version 4 Authentication Dialogue

- *Authentication:* Alice will authenticate to an authentication server (AS) who will provide a temporal permit to demand access for services. This permit is called ticket-granting ticket ($Ticket_{TGS}$) and is comparable to a temporal passport.
- *Access control:* by presenting her $Ticket_{TGS}$ Alice can demand a ticket granting server (TGS) to obtain access for service provided by a specific server $S1$. The TGS decides if the access will be permitted and answers with a service granting ticket $Ticket_{S1}$ for server $S1$.
- *Key exchange:* the authentication server provides a session key for communication between Alice and TGS and the TGS provides a session key for communication between Alice and $S1$. The use of these session keys also serves for authentication purposes.

Figure 2.1 gives an overview of the entities involved in a Kerberos authentication dialogue and the steps of one protocol run:

1. Alice user logs on to her workstation and requests to access a service. From now on, the workstation represents her in the Kerberos protocol and sends the first message to the authentication server AS, containing her name A, the name of an appropriate ticket granting server TGS and a timestamp t_A:

$$A \rightarrow AS: \quad (A, TGS, t_A) \tag{2.13}$$

2. The AS verifies, that A may authenticate herself to access services, generates the key K_A out of Alice's password (which is known to him), extracts the workstation address $Addr_A$ of the request, creates a ticket granting ticket $Ticket_{TGS}$ as well as a session key $K_{A,TGS}$, and sends

the following message to A:

$$AS \rightarrow A: \quad \left\{K_{A,TGS}, TGS, t_{AS}, LifetimeTicket_{TGS}, Ticket_{TGS}\right\}_{K_A} \tag{2.14}$$

with $LifetimeTicket_{TGS}$ defining the maximum time-span in which the ticket is valid and $Ticket_{TGS}$ defined as follows:

$$Ticket_{TGS} = \{K_{A,TGS}, A, Addr_A, TGS, t_{AS}, LifetimeTicket_{TGS}\}_{K_{AS,TGS}} \tag{2.15}$$

3. Upon receipt of this message, the workstation asks Alice to type in her password, computes the key K_A from it, and uses this key to decrypt the message. If Alice does not provide her "authentic" password, the extracted values will be garbage and the rest of the protocol will fail.

   Alice creates a so-called *authenticator* and sends it together with the ticket-granting ticket and the name of server $S1$ to TGS:

   $$A \rightarrow TGS: \quad \left(S1, Ticket_{TGS}, Authenticator_{A,TGS}\right) \tag{2.16}$$

   with $Authenticator_{A,TGS}$ being defined as follows

   $$Authenticator_{A,TGS} = \{A, Addr_A, t'_A\}_{K_{A,TGS}} \tag{2.17}$$

4. Upon receipt, TGS decrypts $Ticket_{TGS}$, extracts the key $K_{A,TGS}$ from it and uses this key to decrypt $Authenticator_{A,TGS}$. If the name and address of the authenticator and the ticket are matching and the time-stamp t_A is still fresh, it checks if A may access the service $S1$, generates a time-stamp t_{TGS}, a session key $K_{A,S1}$ and a $Ticket_{S1}$ for accessing server $S1$, and sends the following message to A:

   $$A \rightarrow TGS: \quad \left\{K_{A,S1}, S1, t_{TGS}, Ticket_{S1}\right\}_{K_{A,TGS}} \tag{2.18}$$

   with $Ticket_{S1}$ being defined as follows:

   $$Ticket_{S1} = \{K_{A,S1}, A, Addr_A, S1, t_{AS}, LifetimeTicket_{S1}\}_{K_{A,S1}} \tag{2.19}$$

5. Alice decrypts the message and does now hold a session key for secure communication with S1. She now sends a message to S1 to show him her ticket and a new authenticator:

   $$A \rightarrow S1: \quad \left(Ticket_{S1}, Authenticator_{A,S1}\right) \tag{2.20}$$

   with $Authenticator_{A,S1}$ being defined as follows:

   $$Authenticator_{A,S1} = \{A, Addr_A, t''_A\}_{K_{A,S1}} \tag{2.21}$$

6. Upon receipt, $S1$ decrypts the ticket with the key $K_{TGS,S1}$ he shares with TGS and obtains the session key $K_{A,S1}$ for secure communication with Alice. Using this key he checks the authenticator and responds to A:

   $$S1 \rightarrow A: \quad \left\{t''_A + 1\right\}_{K_{A,S1}} \tag{2.22}$$

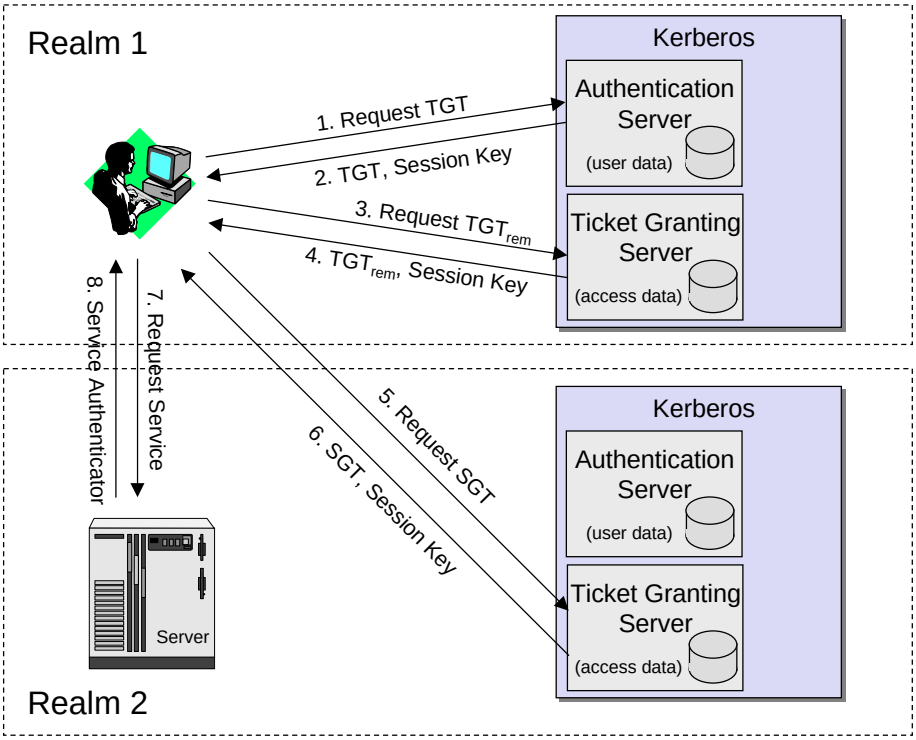


Figure 2.2: Inter-Realm Authentication with Kerberos Version 4

7. By decrypting this message and checking the contained value, Alice can verify, that she is really communicating with $S1$, as only he (besides TGS) knows the key $K_{TGS,S1}$ to decrypt $Ticket_{S1}$ which contains the session key $K_{A,S1}$, and so only he is able to decrypt $Authenticator_{A,S1}$ and to answer with $t''_A + 1$ encrypted with $K_{A,S1}$.

This basic authentication dialogue can be extended to provide multiple-domain authentication. Consider an organization with workstation clusters on two different sites, and imagine that user A of site 1 wants to use a server of site 2: If both sites do use their own Kerberos servers and user databases (containing passwords) then there are in fact two different domains, also called *realms* in Kerberos terminology. In order to avoid that user A has to be registered in both realms, Kerberos allows to perform a so-called *inter-realm authentication*.

Inter-realm authentication requires, that the ticket granting servers of both domains share a secret key $K_{TGS1,TGS2}$. The basic idea is, that the TGS of another realm is viewed as a normal server for which the TGS of the local realm can hand out a ticket. After obtaining the ticket for the remote realm, Alice requests a service granting ticket from the remote TGS (cf. figure 2.2). However, this implies that the remote realm has to trust the Kerberos authentication service of the home domain of a “visiting” user!

The protocol described so far is the Kerberos Version 4 dialogue. A number of deficiencies have been found in this protocol, which can be subdivided into two classes:

- Technical shortcomings: as Kerberos was developped within the project *Athena* and its specific requirements in mind, it did not fully address the need to be of general purpose. This led to the following shortcomings:
  - Encryption system dependence: Kerberos version 4 requires to use the DES encryption algorithm.
  - Internet protocol dependence: the format of the $Addr_A$ field was specified to contain an IP address.
  - Message byte ordering: a proprietary way off specifying the message byte ordering was used.
  - Ticket lifetime: as the ticket lifetime was coded in an eight-bit quantity in units of five minutes, the maximum lifetime was a little over 21 hours, which is not sufficient for some applications, like long running simulations that require a valid ticket granting ticket throughout execution.
  - Authentication forwarding: Kerberos version 4 did not support authentication forwarding which can be quite useful, e.g. giving a print server the permission to access a specific file on behalf of a user.
  - Inter-realm authentication: the scheme for inter-realm authentication does not scale well for a large number N of domains, as $(N^2-N)/2$ shared, secret keys have to be established and maintained in order to allow for complete availability of inter-realm authentication.
- Security deficiencies: apart from the technical limitations mentioned above, there are some security deficiencies in the cryptographic protocol itself:
  - Double encryption: the double encryption of the tickets in the second and fourth step does not provide improved security and is computationally wasteful.
  - PCBC encryption: the non-standard operational mode *propagating cipher block chaining (PCBC)* was used for DES encryption. The intention behind using this mode was to realize confidentiality and data integrity without needing to compute a modification detection code beforehand. Unfortunately, PCBC has been found to be vulnerable to attacks involving the interchange of ciphertext blocks [19].
  - Session keys: each ticket contains a session key, that is used by the client to encrypt an authenticator. The same key may subsequently be used by the client and the server to protect messages exchanged in the course of the service usage. However, as one ticket may be used by a client for multiple, distinct service usages, the service communication of subesequent sessions becomes vulnerable to replay attacks. It would be a better approach to strictly distinguish between keys that are used for authentication and keys that are used for bulk data protection, and provide a means to establish separate keys for the later purpose.
  - Password attacks: as the principal authentication key of each user is derived from the users password, Kerberos is vulnerable to password-guessing attacks. In protocol version 4, the authentication server even answers to the first unprotected message (which might be sent by Alice or any attacker) with a message which is encrypted with this key and which follows a well-known strucure. This makes it even easier for an attacker to launch

a password-guessing attack, as it is relatively easy to obtain a message encrypted with this key which allows to systematically test passwords.

To overcome the above shortcomings, a new version of the Kerberos protocol has been defined [20]. However, this version will not be discussed in detail in this report, as the main objective of this chapter is more to give a brief overview of the concepts of authentication than to explain them thoroughly.

2.2.2 Direct Authentication

As it has been already mentioned above, direct authentication does not require the online participation of a trusted third party. However, if the parties authenticating each other, do not know anything about each other, i.e. they have neither agreed upon a shared, secret key nor do they know the public key of the corresponding party "for sure', some security infrastructure is needed to enable them to establish trust into each other. In the following, a brief overview of the international recommendation X.509 is given, which standardizes a framework for supporting authentication on a global scale.

The ITU-T Recommendation X.509

X.509 is an international recommendation of ITU-T [16] and is part of the X.500-series defining directory services. The first version of X.509 was standardized in 1988, a second version (1993) resolved some security concerns and a third version was drafted in 1995 and standardized in 2000. X.509 defines a framework for provision of authentication services, comprising:

- Certification of public keys and certificate handling:
  - Certificate format,
  - Certificate hierarchy, and
  - Certificate revocation lists.
- Three different dialogues for direct authentication:
  - One-way authentication, requires synchronized clocks,
  - Two-way mutual authentication, still requires synchronized clocks, and
  - Three-way mutual authentication entirely based on random numbers.

Certification of Public Keys
The main motivation behind X.509's recommendations for public key certificates is to ensure the authenticity of public keys. A public key certificate is some sort of passport, certifying that a public key belongs to a specific name, in the context of X.509 this means specifically an X.500 name. Certificates are issued by so-called *certification authorities (CA)*. In a public key certificate the issuing CA signs with her private key, that the public key contained in the certificate belongs to the subject name stated in the certificate.

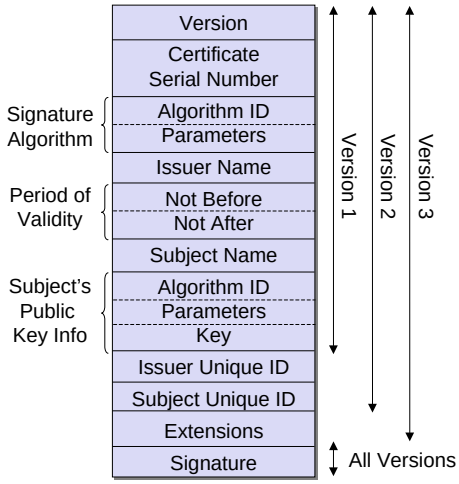


Figure 2.3: The X.509 Certificate Format

If all potential communication partners of the entity, say Alice, named in a certificate know *for sure* the public key of the issuing CA, they are able check the certificates (including Alice's) issued by this CA. So, if we consider a population of N entities the problem of "authentically" distributing all N public keys to all entities of the population can be reduced to "authentically" distributing the public certification key $+CK_{CA}$ of the CA to all entities. In order to learn about the authenticity of an unknown public key, every entity can check the corresponding public key certificate, issued by the CA. Public key certificates can avoid online-participation of a TTP, as these certificates do not need to be kept confidential and they can be made publicly available, X.509 recommends for this purpose the public directory specified in the X.500 series of recommendations. However, the security of the private key of the CA is crucial to the security of all users, as the compromise of this key allows to forge certificates. Figure 2.3 shows the format of an X.509 certificate. A shorthand notation for a public certificate $Cert_{-CK_{CA}}(+K_A)$ issued by CA and certifying the authenticity of A's public key $+K_A$ that is commonly used is $CA << A >>$.

As the deployment of one single CA is not desirable for very large populations, X.509 specifies a method to *chain* certificates by allowing CAs to certify the public keys of other CAs. Consider two entities Alice (A) and Bob (B) and their corresponding certification authorities CA and CB. If both CA and CB certify each others public key with certificates $CA << CB >>$ and $CB << CA >>$, then Alice can verify Bob's public key by checking the *certificate chain* $CA << CB >>,\ CB << B >>$ and Bob can verify Alice's public key by checking the certificate chain $CB << CA >>,\ CA << A >>$.

The possible length of certificate chains is not restricted to two, which allows establish a trust relationship between two entities by following a *chain of trust* between multiple CAs. However, as in the general case it might not be obvious to decide which certificates have to be checked in order to establish a chain of trust between the CAs of two entities A and B, X.509 recommends to arrange certification authoroties in a so-called *certification hierarchy*, so that it is straightforward to decide which certificates have to be retrieved from the public directory.

Figure 2.4 shows a hypothetical certification hierarchy. If, for example, entity A would like to verify

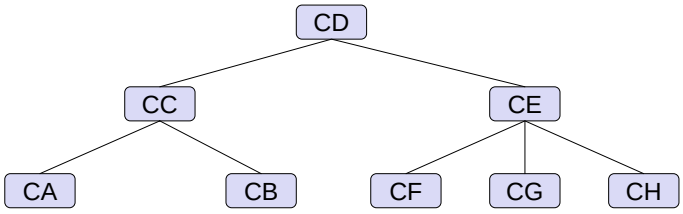


Figure 2.4: A Hypothetical Certification Hierarchy

the authenticity of the public key of entity G, it would have to check the certificate chain:

$$CA << CC >>,\ CC << CD >>,\ CD << CE >>,\ CE << CG >>,\ CG << G >> \tag{2.23}$$

If the private key of an entity Alice is ever compromised, e.g. because Eve broke into her computer, read her private key from a file, and cracked the password Alice used to protect the private key, the corresponding public key should be publicly invalidated as soon as the compromise is discovered. This is done by *revoking* the public key certificate. If the certificate would not be revoked, then Eve could continue to impersonate Alice up to the end of the certificate's validity period. An even worse situation occurs, when the private key of a certification authority is compromised, as this implies, that all certificates signed with this key have to be revoked!

Certificate revocation is realized by maintaining *certificate revocation lists (CRL)*. CRLs are stored in the X.500 directory and when checking a certificate, an entity also has to check that the certificate has not yet been revoked which is realized by searching for the certificate in the CRL. Certificate revocation is a relatively slow and expensive operation, as the revocation information has to be distributed with a public directory.

Direct Authentication Protocols

The X.509 recommendation also defines a family of three direct authentication protocols which are succesively based on each other and can be executed by arbitrary entities:

1. *One-way authentication:* If only Alice wants to authenticate herself to Bob she sends the following message to Bob:

$$A \rightarrow B\colon\ \big(A[t_A, r_A, B, sgnData_A, \{K_{A,B}\}_{+K_B}],\ CA << A >>\ \big) \tag{2.24}$$

   with $sgnData_A$ representing optional data to be signed by A, $\{K_{AB}\}_{+K_B}$ being an optional session key encrypted with Bob's public key, and $CA << A >>$ being optional as well. The notation $A[m]$ serves as a shorthand notation for $(m, \{MDC(m)\}_{-K_A})$. Upon reception of this message, Bob verifies with $+CK_{CA}$ the contained certificate, extracts Alice's public key, checks Alice's signature of the message and the timeliness of the message (by comparing t_A to its own clock), and optionally decrypts the contained session key $K_{A,B}$, Alice has proposed.

2. *Two-way authentication:* If mutual authentication is desired, then Bob creates a similar message:

$$B \rightarrow A\colon\ \big(B[t_B, r_B, A, r_A, sgnData_B, \{K_{B,A}\}_{+K_A}],\ CA << B >>\ \big) \tag{2.25}$$

The contained timestamp t_B is not really required, as Alice can verify the freshness of the signed message by checking if it contains the random number r_A.

3. *Three-way authentication:* If Alice and Bob are not sure if they have synchronously running clocks, Alice sends the following message to Bob:

$$A \rightarrow B: \quad \big(A[r_B]\big) \tag{2.26}$$

In this case, the timeliness of Alice's participation in the authentication dialogue is proven by signing the "fresh" random number r_B.

Concerning the signature algorithm it can be remarked, that as obvious from the use of certificates, X.509 suggests to sign the authentication messages using asymmetric cryptography. However, the authentication protocol itself can also be deployed using symmetric cryptography. In this case, A and B need to have agreed upon a secret authentication key $AK_{A,B}$ prior to any protocol run, and the messages are signed by appending a MAC computed with that key, such that $A[m]$ represents $(m, \{MDC(m)\}_{K_{A,B}})$.

2.2.3 Validation of Cryptographic Protocols

The literature on cryptographic protocols gives various examples of weak protocols, such that an attacker could circumvent the protocol without possessing the necessary key(s) or breaking the cryptographic algorithm used in the protocol [26]. Examples are the *Needham-Schroeder protocol* [32], in which an attacker could present an old session key and use it for a new authenticated session [8] (see also the discussion above), the authentication protocol of an early draft version of the international standard *X.509* [15] which contained a similar flaw [6], as well as a *software licensing system of Purdy, Simmons and Studier* [36], which could be circumvented by an attacker by combining and reusing recorded messages [40].

These examples show the need for formal validation of cryptographic protocols as protocol flaws can not be sufficiently analyzed using non-formal methods. For this purpose a variety of approaches has been developed that can be divided into the following four classes [25]:

1. *General approaches for analysis of specific protocol properties:* Cryptographic protocols are analyzed using established methods of software-verification, like finite-state-machine based approaches [39, 54], first-order predicate calculus [17], or use specification languages for description and analysis of cryptographic protocols [55]. However, reasoning about the security of a cryptographic protocol differs significantly from the proof of correctness of a protocol, as the latter does not have to take into account malicious manipulations. Thus the approaches of this category are not sufficiently suited for analysis of attacks on cryptographic protocols.

2. *Expert system based approaches:* The knowledge of human experts is formalized into deductive rules that can be used by a protocol designer to investigate different scenarios in an automated or even interactive way [23, 29]. While this approach is well suited to analyze a protocol's resistance to known attacks it does not allow to find flaws in a protocol that are based on unknown attacking techniques [38, p. 66].

3. *Algebraic approaches:* Cryptographic protocols are specified as algebraic systems whereby in addition to the protocol steps, also the peer entities' knowledge and beliefs concerning the authentication dialogue are included in the formal model. The analysis of the resulting model is conducted by examining algebraic term-rewriting properties of the model and inspecting if the model can attain certain desirable or undesirable states. Examples for approaches of this class are [28, 50, 51, 52, 56].

4. *Specific logic based approaches:* Approaches of this class define a set of predicates and a mapping of messages exchanged during a protocol run into to a set of formula. A generic set of rules allows then to analyze the *knowledge* and *belief* that is obtained by the peer entities of a cryptographic protocol during a protocol run. The first published approach of this class was *BAN Logic* [6], named after its inventors Burrows, Abadi and Needham. Various extensions and other approaches based on the same idea have been proposed since then [10, 12, 18, 24, 41, 53]. Other logics based validation techniques for cryptographic protocols are [3, 4, 30, 37, 43, 44, 45, 46, 47].

   One of the most successful approaches of this category is *GNY Logic*, which has been widely used to analyze cryptographic protocols since its publication [12].

2.3 Conclusion

This chapter gave a brief introduction into principles of authentication. While *data origin authentication* aims to ensure the integrity of messages and to provide assurance that the identity claiming to have created a message is indeed the originator of the message, *entity authentication* additionally protects against replay attacks, ensuring that the peers of a communication taking place are actually participating in the communication at a given moment.

The cryptographic algorithms used in authentication exchanges are *symmetric* and *asymmetric encryption* as well as *cryptographic hash functions*. These algorithms are used as base primitives in building *cryptographic protocols* for specific *security objectives*.

Authentication protocols are an important class of cryptographic protocols. Depending on if an authentication protocol comprises online participation of a *trusted third party* or not, the authentication protocol is referred to as an *arbitrated authentication* or a *direct authentication* protocol. Authentication protocols are fragile in the sense that a small change in message contents or order can break the security of an authentication protocol. Therefore, formal analysis should be conducted when designing a new or modifying an existing authentication protocol.

Chapter 3

Current Approaches to Authentication in Mobile Communications

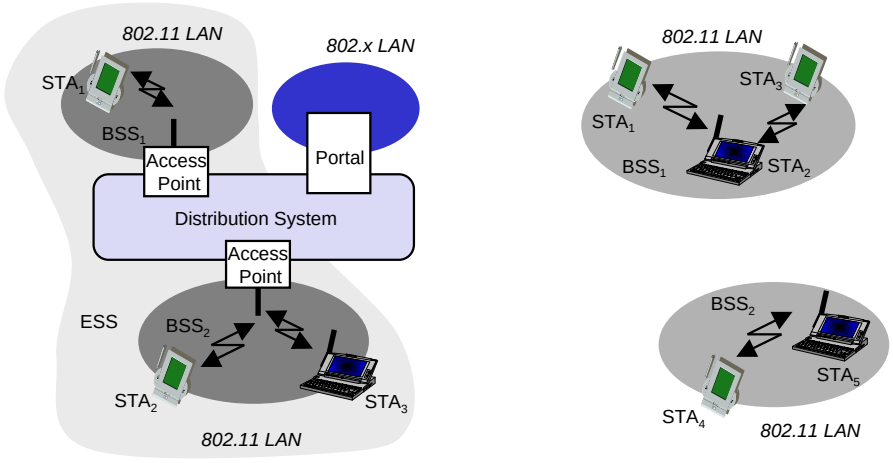
This chapter analyzes the authentication procedures of current wireless and mobile communications standards. The next section describes the security mechanisms including authentication of IEEE 802.11, the most deployed wireless LAN standard today. While its mobility support is limited as a result of its conception of a local area network, it may nevertheless serve as an underlying link-layer technology in a true mobile communications network, e.g. based on Mobile IP. Therefore, authentication in IEEE 802.11 is included in this project's analysis. Section 3.2 describes the authentication protocol of the global standard for mobile telephony, *Global System for Mobile Communications (GSM)* and section 3.3 describes authentication of the so-called *Release '99* of the new standard *Universal Mobile Telecommunications System (UMTS)* which is basically a further development of GSM authentication. Authentication for Mobile IP is discussed in section 3.4 with one section describing the standard Mobile IP registration authentication and one section devoted to newer standardization efforts in the IETF regarding Mobile IP and AAA interworking. The chapter closes with a conclusion comparing the benefits and deficiencies of the different approaches.

3.1 Authentication in IEEE 802.11

The most deployed standard today for wireless local area networks, IEEE 802.11 [14], standardizes medium access control and physical characteristics of a wireless local area network (LAN). The standard comprises three physical layer units:

- Frequency Hop Spread Spectrum: 2.4 GHz band; 1, 2, 5.5, and 11 Mbit/s
- Direct Sequence Spread Spectrum: 2.4 GHz band, 1, 2, 5.5, 11 and 22 Mbit/s
- Baseband infrared: diffuse infrared; 1 and 2 Mbit/s

Transmission in the license-free 2.4 GHz band implies that the medium has to be shared with unvolunteering 802.11 devices, and that logical separated wireless LANs will geographically and phys-



a.) Infrastructure Network

b.) Ad-Hoc Network

Figure 3.1: Architecture of IEEE 802.11 Networks

ically overlap. The medium access control of IEEE 802.11 supports operation under control of an access point as well as between independent stations. Figure 3.1 illustrates the architectures of both modes of operation and makes use of the following acronyms:

- *Station (STA):* terminal with access mechanisms to the wireless medium and radio contact to the access point,
- *Basic Service Set (BSS):* group of stations using the same radio frequency,
- *Access Point:* station integrated into the wireless LAN and the distribution system,
- *Portal:* bridge to other (wired) networks,
- *Distribution System:* interconnection network to form one logical network *(extended service set, ESS)* based on several BSS.

IEEE 802.11 provides two basic security services:

- authentication of 802.11 peer entities, e.g. a mobile station and an access point, and
- confidentiality of data transfer, this is referred to as *wireless equivalent privacy (WEP)* in 802.11 terminology.

Many vendors claim that IEEE 802.11 is as secure as a wired network, which is also intended to be indicated by the name wired equivalent privacy. However, this is far from being true, as quite a few security flaws have been found in the specification [5] with the worst drawback being the missing key management resulting in the shared use of one static key per basic service set.

IEEE 802.11 authentication should be performed between stations and access points and could also be performed between arbitrary stations. When performing authentication, one station is acting as the requestor (A) and the other one as the responder (B). The authentication dialogue proceeds as follows:

1. The requestor A sends a message demanding to authenticate itself to the responder and containing the command identifier 1 and his identity Id_A:

$$A \rightarrow B: \quad \big(Authentication, 1, Id_A\big) \tag{3.1}$$

2. Upon reception of this message the responder generates a fresh random number r_B and answers with the following message:

$$B \rightarrow A: \quad \big(Authentication, 2, r_B\big) \tag{3.2}$$

3. The requestor creates the third message which contains the random number generated by B and is encrypted with the shared, secret key K_{BSS} that constitutes the shared secret of the basic serving set in which A and B communicate:

$$A \rightarrow B: \quad \big\{Authentication, 3, r_B\big\}_{K_{BSS}} \tag{3.3}$$

4. Upon reception of this message, the responder decrypts it with K_{BSS} and checks if the contained number is in fact the random number he generated in step 2. If this is the case, B answers with a positive response:

$$B \rightarrow A: \quad \big(Authentication, 4, Succesful\big) \tag{3.4}$$

As can be easily deduced from the above protocol, mutual authentication requires two independent protocol runs, one in each direction. It has to be noted, that it is not possible to explicitely demand an 802.11 device to authenticate itself to another device, as the dialogue has to be initiated by the entity requesting to authenticate itself to some other device.

IEEE 802.11 provides two "variants" of authentication:

- *Open System Authentication:* "essentially it is a null authentication algorithm" ([14, section 8.1.1]), that means authentication is turned off, and
- *Shared Key Authentication:* "Shared key authentication supports authentication of STAs as either a member of those who know a shared secret key or a member of those who do not." ([14, section 8.1.2])

Furthermore, the required shared and secret key is presumed to have been delivered to participating STAs via a secure channel that is independent of IEEE 802.11. Concluding, IEEE 802.11 does not provide sufficient means for authentication in truly mobile environments and as a result of the missing key management very often "open system authentication" is used.

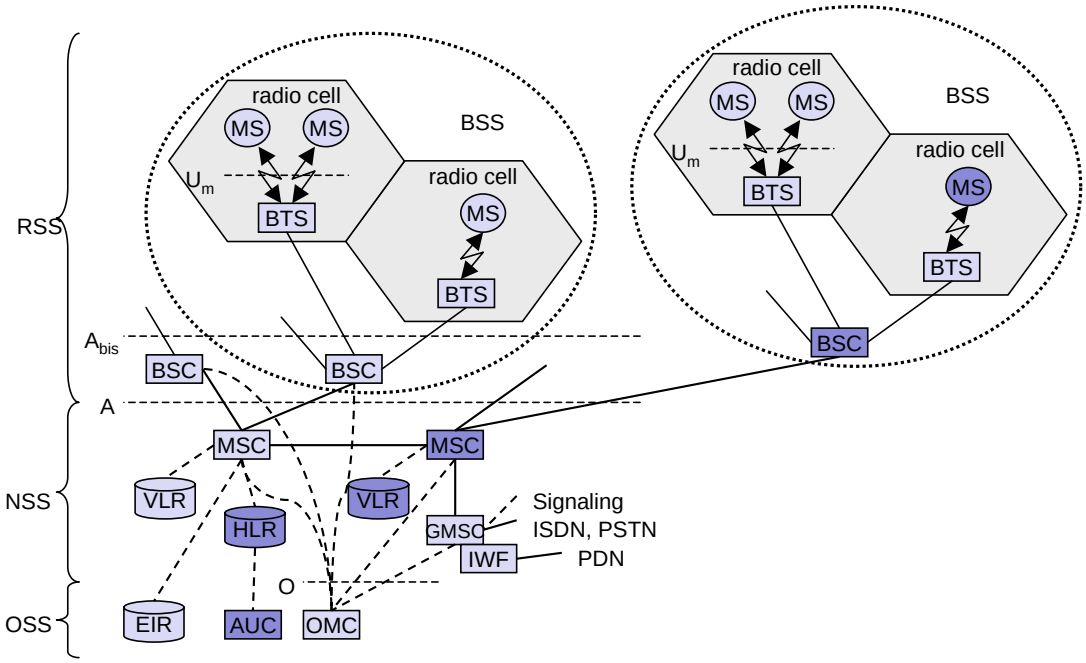


Figure 3.2: GSM Architecture and Entities Involved in Authentication

3.2 Authentication in GSM

The pan-european standard *Global System for Mobile Telecommunications (GSM)* [48, 49] aims to provide true mobile, wireless communications with support for voice and data services and allows for worldwide connectivity and international mobility with unique addresses. GSM provides the following security features:

- *Subscriber identity confidentiality:* provides protection against an intruder trying to identify which subscriber is using a given resource on the radio path (e.g. traffic channel or signaling resources) by listening to the signaling exchanges on the radio path,
- *Subscriber identity authentication:* protects the network against unauthorized use,
- *Signaling information element confidentiality:* aims to ensure non-disclosure of signaling data on the radio link, and
- *User data confidentiality:* aims to ensure non-disclosure of user data on the radio link.

However, only eavesdropping attacks on the radio link between the mobile and the base stations have been taken into account in the design of the confidentiality mechanisms and no security is provided inside the fixed part of the network. In the following we will focus on the subscriber identitiy authentication of GSM. Figure 3.2 illustrates the architecture of an GSM network with respect to the entities involved in user authentication and table 3.1 lists common acronyms of the GSM standards.

Table 3.1: Common GSM Acronyms

Acronym	Meaning
AUC	Authentication center
BSC	Base station controller
BTS	Base transceiver station
IMSI	International mobile subscriber identity
HLR	Home location register
LAI	Location area identifier
MS	Mobile station (e.g. a mobile phone)
MSC	Mobile switching center
MSISDN	Mobile subscriber international ISDN number
SIM	Subscriber identity module
TMSI	Temporary mobile subscriber identity
VLR	Visitor location register

The GSM subscriber identity authentication is realized with a challenge-response dialog. For this the *subscriber identity module (SIM)* and the *authentication center (AUC)* of the subscribers network provider share a secret key $K_{AUC,MS}$. In order prepare user authentication, the AUC generates a vector of random numbers $R_{AUC:1,n}$ and uses two algorithms called A3 and A8 to generate two vectors of expected responses $SRES_{AUC:1,n}$ and session keys $K_{BSC,MS:1,n}$. Together the three vectors form an authentication vector which is stored in the *home location register (HLR)* storing the current location of the *mobile station (MS)*.

When a mobile station needs to authenticate to a serving network, it may be in one of the following situations:

- The current cell belongs to a network, the MS has not visited in the (near) past. In this case it presents his *international mobile subscriber identity (IMSI)* to the serving network. The serving network, e.g. its *mobile switching center (MSC)* determines and asks the appropriate HLR to send an authentication vector which is stored in the *visited location register (VLR)* together with the IMSI of the mobile station.
- The current cell belongs to a network to which the MS has already authenticated in the (near) past. If the authentication vector of the mobile station is still available in the VLR and there are still some triplets left, that have not yet been used, then the HLR of that mobile station needs not to be contacted.

In both cases an unused[1] random number $R_{AUC:i}$ is presented to the mobile station and the station answers this challenge with the expected result $SRES_{AUC:i}$ that it computes with the algorithm A3 and the key $K_{AUC,MS}$ it shares with its authentication center AUC. Figure 3.3 illustrates the basic scheme of GSM subscriber identity authentication.

[1]The GSM standards also allow to reuse authentication triplets as an implementation specific choice. However, this introduces the risk of not being able to detect potential replay attacks.

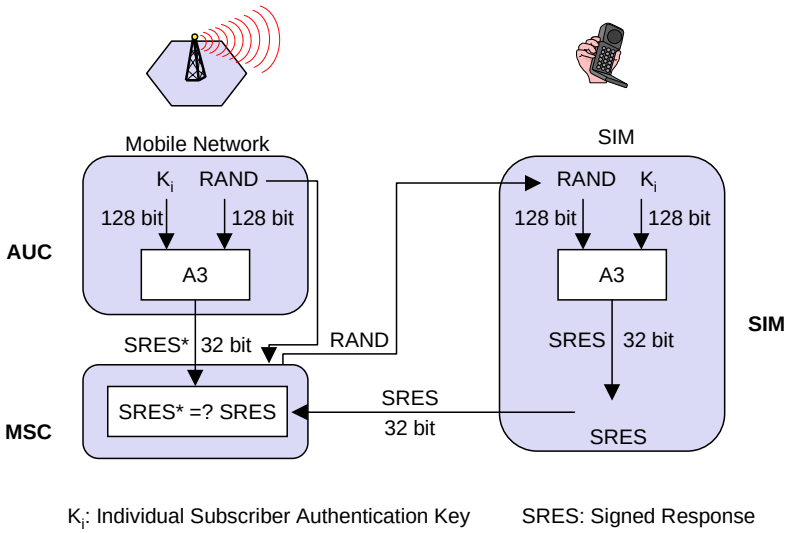


Figure 3.3: Basic GSM Authentication Scheme

The actual protocol of a “first-time” authentication consists of the following steps:

1. The mobile station sends its international mobile subscriber identity to the base station of the visited network:

$$MS \rightarrow BSC: \quad \left(IMSI_{MS}\right) \tag{3.5}$$

2. The BSC sends this information to its MSC which determines with a lookup in the VLR that a new authentication vector for this MS has to be obtained from the HLR of that MS. So, in short the visited network (represented by the BSC in the formula) sends the $IMSI_{MS}$ to the appropriate HLR in order to ask for a new authentication vector:

$$BSC \rightarrow HLR: \quad \left(IMSI_{MS}\right) \tag{3.6}$$

3. Upon receipt of this message the HLR looks up an authentication vector belonging to $ISMI_{MS}$ (which may be generated on the fly if the HLR is also acting as an AUC or has an AUC directly attached to it) and sends it back to the visited network:

$$HLR \rightarrow BSC: \quad \left(IMSI_{MS}, K_{BSC,MS:1,n}, R_{AUC:1,n}, SRES_{AUC:1,n}\right) \tag{3.7}$$

4. The visited network then sends a challenge containing one of the random numbers received in the authentication vector to the mobile station:

$$BSC \rightarrow MS: \quad \left(R_{AUC:1}\right) \tag{3.8}$$

5. Upon receipt of the challenge the mobile station computes the signed response $SRES_{AUC:1} = A3(K_{AUC,MS}, R_{AUC:1})$ and sends it back to the visited network:

$$MS \rightarrow BSC: \quad \left(SRES_{AUC:1}\right) \tag{3.9}$$

6. The visited network checks if the response sent by the mobile station equals the number that has been provided in the authentication vector, and if so extracts the appropriate session key $K_{BSC,MS:1}$ from the authentication vector. It then creates a message containing the *location area identifier* LAI_1 that identifies the area where the MS currently is located and a *temporary mobile subscriber identity (TMSI)* for the MS. This message is encrypted with the session key before being sent to the mobile station:

$$BSC \rightarrow MS: \quad \left\{LAI_1, TMSI_{MS:1}\right\}_{K_{BSC,MS:1}} \tag{3.10}$$

7. The MS generates the session key by computing $K_{BSC,MS:1} = A8(K_{AUC,MS}, R_{AUC:1})$ and uses it to encrypt the received message. After decryption the MS stores the values LAI_1 and $TMSI_{MS:1}$. This temporary identity is used in all further signaling exchanges so that the $IMSI_{MS}$ needs not to be exposed to potential attackers eavesdropping on the air interface.

If the MS later on needs to re-authenticate in the range of the same VLR, e.g. in the course of a handover to another BSC, this is accomplished as follows:

1. The MS sends a message containing the location area identifier and its temporary mobile subscriber identity:

$$MS \rightarrow BSC: \quad \left(LAI_1, TMSI_{MS:i}\right) \tag{3.11}$$

2. Upon receipt of this message the BSC checks (via the MSC) with the VLR if this temporary identity is known in this area and obtains an unused triplet of the authentication vector. It then sends the following challenge to the MS:

$$BSC \rightarrow MS: \quad \left(R_{AUC:i+1}\right) \tag{3.12}$$

3. The MS computes the signed response $SRES_{AUC:i+1} = A3(K_{AUC,MS}, R_{AUC:i+1})$ and sends it back to the BSC:

$$MS \rightarrow BSC: \quad \left(SRES_{AUC:i+1}\right) \tag{3.13}$$

4. The BSC checks if the returned response matches the expected result of the authentication triplet. If so, it generates a new temporary mobile subscriber identity, encrypts it and the location area identifier with the new session key $K_{BSC,MS:i+1}$, and sends the resulting message to the MS:

$$BSC \rightarrow MS: \quad \left\{LAI_1, TMSI_{MS:i+1}\right\}_{K_{BSC,MS:i+1}} \tag{3.14}$$

5. The MS computes the new session key $K_{BSC,MS:i+1} = A8(K_{AUC,MS}, R_{AUC:i+1})$ and uses it to obtain its new temporary identity $TMSI_{MS:i+1}$ that will be used in future signaling and authentication message exchanges. As the temporary identity is send to the mobile station in an encrypted way, a passive eavesdropper can not link the new temporary identity to the former one.

A similar procedure is used in the case when a mobile station enters an area which is handled by another VLR. If the two VLRs belong to the same operator, the new VLR_2 may determine the old VLR_1 with the help of the location area identifier LAI_1 and then ask VLR_1 to send remaining authentication triplets for this mobile station.

1. The MS sends a message containing the location area identifier and its temporary mobile subscriber identity:

$$MS \rightarrow BSC: \quad \left(LAI_1, TMSI_{MS:i}\right) \tag{3.15}$$

2. Upon receipt of this message the BSC checks (via the MSC) with the VLR_2 if this temporary identity is known in this area. As LAI_1 contained in the mobile stations request does not match the local LAI_2 the local VLR_2 contacts VLR_1 in order to ask for authentication triplets for that mobile station:

$$VLR_2 \rightarrow VLR_1: \quad \left(LAI_1, TMSI_{MS:i}\right) \tag{3.16}$$

3. Upon receipt of this message VLR_1 looks up the entry corresponding to the temporary pseudonym and answers with a message containing the temporary pseudonym and the IMSI of the mobile station as well as the remaining unused authentication triplets:

$$\begin{aligned} VLR_1 \rightarrow VLR_2: \quad & \big(LAI_1, TMSI_{MS:i}, IMSI_{MS}, \\ & K_{BSC,MS:i+1,n}, R_{AUC:i+1,n}, SRES_{AUC:i+1,n}\big) \end{aligned} \tag{3.17}$$

4. After receiving this message VLR_2 communicates an authentication triplet to the BSC which in turn sends a challenge to the mobile station:

$$BSC \rightarrow MS: \quad \left(R_{AUC:i+1}\right) \tag{3.18}$$

5. The MS computes the signed response $SRES_{AUC:i+1} = A3(K_{AUC,MS}, R_{AUC:i+1})$ and sends it back to the BSC:

$$MS \rightarrow BSC: \quad \left(SRES_{AUC:i+1}\right) \tag{3.19}$$

6. The BSC checks if the returned response matches the expected result of the authentication triplet. If so, it generates a new temporary mobile subscriber identity, encrypts this and its own location area identifier LAI_2 with the new session key $K_{BSC,MS:i+1}$, and sends the resulting message to the MS:

$$BSC \rightarrow MS: \quad \left\{LAI_2, TMSI_{MS:i+1}\right\}_{K_{BSC,MS:i+1}} \tag{3.20}$$

7. The MS computes the new session key $K_{BSC,MS:i+1} = A8(K_{AUC,MS}, R_{AUC:i+1})$ and uses it to obtain its new temporary identity $TMSI_{MS:i+1}$ and the new location area identifier which will be used in future signaling and authentication message exchanges.

While the scheme explained above is the preferred one, it can not be used in certain situations, e.g.:

- the $TMSI_{MS:i}$ is unavailable at VLR_1, e.g. because it has not been used for a longer amount of time,
- there are no more unused authentication triplets left, or
- VLR_2 is not able to contact VLR_1.

In these cases an initial dialogue is needed, that is the BSC asks the mobile station to send its IMSI and contacts the HLR of the mobile station in order to ask for a new authentication vector. Furthermore, if VLR_1 and $VL!R_2$ belong to different network operators the handover can not be performed and the call is disconnected.

Summarizing, in GSM only the mobile station authenticates itself to the visited network and there is no authentication of the visited network to the mobile station. The authentication is based on challenge-response vectors which are generated by an authentication center of the mobile stations operator and which are transmitted unprotected via the signaling network to the visited network. This allows for two main attacks in case the signaling network is compromised:

- The most obvious attack is an attacker eavesdropping on a signaling link in order to obtain valid challenge-response vectors and IMSIs. This would allow to access service on behalf of other users.
- As visited network has no assurance of the freshness of the received authentication vector, an active attacker could replay an old vector. Even worse, as the visited network has no means to check the authenticity of received authentication vectors, an attacker could even invent them. As a result the attacker could access service on behalf of other users and / or “invent” new users.

The permanent identification of a mobile station (IMSI) is just sent over the radio link when no temporary pseudonym and / or authentication vector is available in the visited network for that mobile. This allows for partial location privacy. However, as the IMSI is sometimes sent in clear, it is nevertheless possible to learn about the location of those mobile stations who are performing a “first time” authentication by eavesdropping on the radio link. An active attacker may even impersonate a base station and explicitly demand mobile stations to send their IMSIs, so that the location privacy protection of GSM can not be considered sufficient. Finally, the trust model of GSM which assumes trust between all network operators can not be considered adequate for future generations of wireless networks that are supposed to support seamless handover between a variety of network technologies including privatetly operated wireless local area networks and publicly operated wide area mobile networks (UMTS and beyond).

3.3 Authentication in UMTS Release '99

The *release '99* of the UMTS specification lists the following security services to be provided:

- User identity confidentiality:
  - *User identity confidentiality:* the property that the permanent user identity (IMSI) of a user to whom a service is delivered cannot be eavesdropped on the radio access link,
  - *User location confidentiality:* the property that the presence or the arrival of a user in a certain area cannot be determined by eavesdropping on the radio access link,
  - *User untraceability:* the property that an intruder cannot deduce whether different services are delivered to the same user by eavesdropping on the radio access link,
- Entity authentication:
  - *User authentication:* the property that the *serving network (SN)* corroborates the user identity of the user,
  - *Network authentication:* the property that the user corroborates that he is connected to a serving network that is authorized by the user's HE to provide him services; this includes the guarantee that this authorization is recent,
- Confidentiality:
  - *Cipher algorithm agreement:* the property that the mobile station (MS) and the SN can securely negotiate the algorithm that they shall use subsequently,
  - *Cipher key agreement:* the property that the MS and the SN agree on a cipher key that they may use subsequently,
  - *Confidentiality of user data:* the property that user data cannot be eavesdropped on the radio access interface,
  - *Confidentiality of signaling data:* the property that signaling data cannot be eavesdropped on the radio access interface,
- Data Integrity:
  - *Integrity algorithm agreement*,
  - *Integrity key agreement*,
  - *Data integrity and origin authentication of signaling data:* the property that the receiving entity (MS or SN) is able to verify that signaling data has not been modified in an unauthorized way since it was sent by the sending entity (SN or MS) and that the data origin of the signaling data received is indeed the one claimed.

Table 3.2: Common UMTS Acronyms

Acronym	Meaning
AK	Anonymity key
AMF	Authentication management field
AUTN	Authentication token
AV	Authentication vector
CK	Cipher key
HE	Home environment
IK	Integrity key
RAND	Random challenge
SQN	Sequence number
SN	Serving network
USIM	User services identity module
XRES	Expected response

Table 3.2 lists common UMTS acronyms with special regard to security mechanisms. Authentication in UMTS basically follows the same ideas like in GSM. Some entity in the home network generates an authentication vector for a mobile station, which are send to the visited network. One difference to GSM, however, is the fact that the home network also provides some values that enable the mobile station to verify if it is receiving responses from a visited network that his home network provider trusts in.

Figure 3.4 gives an overview of the UMTS authentication procedure:

- After the mobile station has presented its identity to the visited network, the visited network requests authentication data from the stations home environment.
- After receiving and storing an appropriate authentication vector the visited network sends the first random number of the authentication vector together with a so-called *authentication token (AUTN)* to the mobile station.
- The mobile station checks the authentication token, computes the response to the random number challenge and sends it back to the visited network. Furthermore, the mobile station computes the cipher key and the integrity key from the random number and the secret key it shares with its home environment.
- The visited network checks, if the mobile stations response matches the expected response of the authentication vector and if so selects the cipher key and the integrity key from the authentication vector.

Figure 3.5 illustrates the generation of the authentication vector in the home environment:

- The home environment remembers a sequence number SN for each of its mobile stations. For

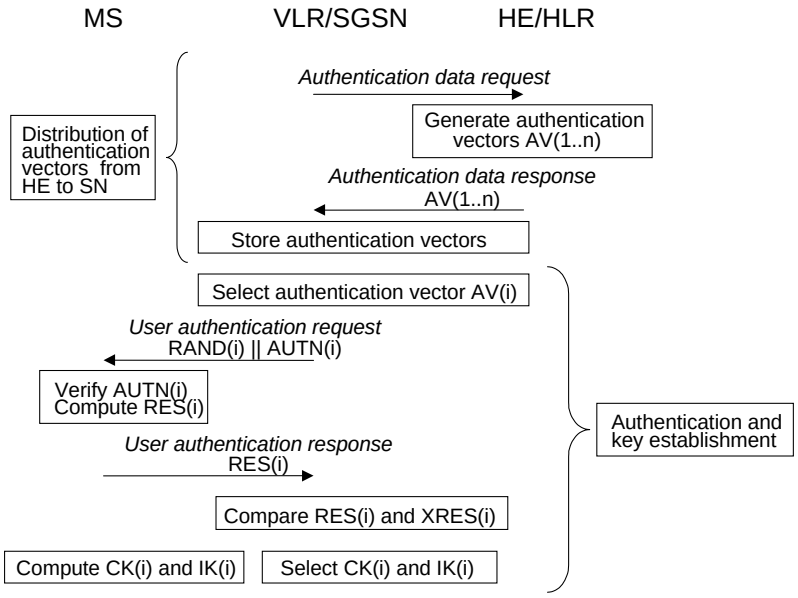


Figure 3.4: Overview of Authentication in UMTS Release '99 (Source [1])

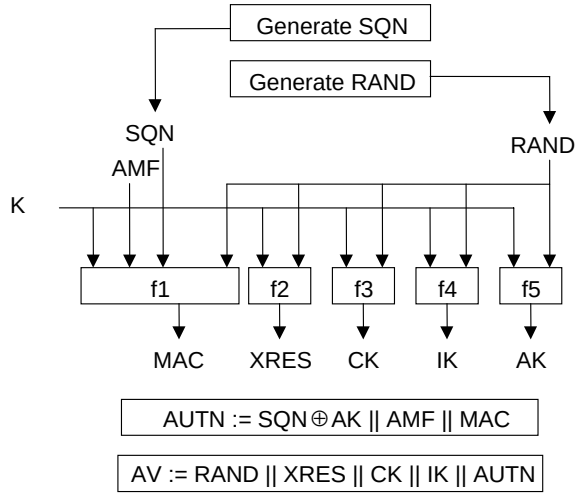


Figure 3.5: Generation of Authentication Vectors in UMTS Release '99 (Source [1])

each authentication entry of the authentication vector it generates a new sequence number (by incrementing the stored value) and a fresh random value $RAND$.

- By computing a function $f1$ over the key K shared with the mobile station, s so-called *authentication management field (AMF)*, the sequence number SN and the random number $RAND$ it generates a a so-called $message authentication code (MAC)$[2].

[2]This code is not to be mixed up with the definition of the term MAC given in section 2.1

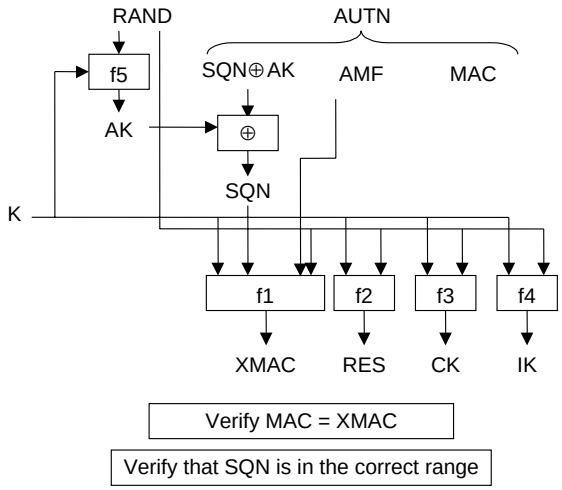


Figure 3.6: User Authentication Function in the User Service Identity Module (Source [1])

- Like in GSM the expected result is computed by applying a function to the shared secret key and the random number, and additionally three different keys are computed in a similar fashion, each one with a different function that is applied to the shared secret key and the random number: a ciphering key (CK), an integrity key (IK) and a so-called *anonymity key* (AK).

- The purpose of the anonymity key is to conceal the sequence number, which is communicated to the mobile station in the authentication token. This token is composed of the sequence number xor'ed with the anonymity key, the authentication management field, and the MAC.

- An authentication vector is made up of entries composed of a random number RAND, an expected result XRES, a cipher key CK, an integrity key IK, and an authentication token.

When the mobile station receives the challenge it proceeds as follows (see also figure 3.6):

- It computes the anonymity key by applying the function $f5$ to the shared secret key and the random number.

- By xor'ing the anonymity key with the concealed sequence number, it obtains the sequence number of this authentication challenge.

- It then computes the MAC by applying the function $f1$ to the shared secret key, the sequence number, the authentication management field, and the random number. If the computed MAC matches the MAC contained in the authentication token and the sequence number is bigger than the sequence number of the last successful protocol run, the mobile station assumes that the response is from a visited network in which its home environment trusts.

- After the network authentication has been checked the mobile station compute the cipher key, the integrity key and the expected result, and sends the latter to the visited network.

Summarizing, security of UMTS Release '99 is quite similar to its GSM counterpart. The home environment generates challenge-response vectors which are transmitted unprotected via the signaling network to a visited network that needs to check the authenticity of a mobile station. However, unlike in GSM the network also authenticates itself to the mobile station. The permanent identification of a user is still revealed to the visited network and is only protected from passive eavesdropping attacks on the wireless link in subsequent authentication and signaling exchanges. Nevertheless, it can still be demanded by an attacker which impersonates a base station, as there is no network authentication in this case. The reason for this is, that the permanent identification of the mobile station is needed in order to demand the home environment to provide an authentication token for the visited network. Furthermore, confidentiality is only provided on the radio link and the the security model still assumes trust between all network operators. Finally, no security measures are taken to protect security relevant information during transport in the signaling network.

3.4 Authentication for Mobile IP

When reasoning about and designing an authentication infrastructure and protocol for Mobile IP it should be taken into account, that there are different motivations for different authentication relations:

- *Authentication between the mobile node and its home network* basically serves to counter *hijacking attacks*, which may enable a malicious node to obtain access to the IP packets destined for a mobile node.
- *Authentication between the mobile node and the visited network* serves to be able to control access to network resources and to ensure secure accounting of network resource usage.
- *Authentication between the visited network and the home network* also serves to control which mobile node may use network resources and to ensure secure accounting of network resource usage. Additionally it allows to control which networks may be accessed by a mobile node.

Even though Mobile IP provides means to realize all of the above relations it has soon been discovered, that the build-in mechanisms are not sufficient to realize scalable authentication in roaming scenarious on a global basis. The main reasons for this lie in the missing key management and the missing integration with an authorization and accounting infrastructure. Therefore, two working groups of the IETF, the AAA group and the Mobile IP group, are currently analyzing the interactions between Mobile IP authentication and AAA procedures in a joint effort.

The following section describes the standard Mobile IP authentication protocol and section 3.4.2 gives an overview of the current state of integrated AAA / Mobile IP authentication.

3.4.1 Standard Mobile IP Authentication

Mobile IP provides basic mechanisms for authenticating the entities involved in a mobile nodes registration. The principle mechanism for realizing authentication is appending a cryptographic hash value to registration messages. These hash values are transmitted in extensions to the registration messages.

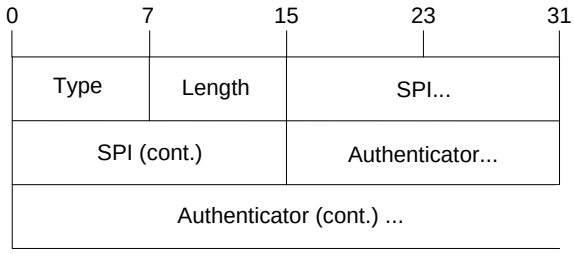


Figure 3.7: Mobile IP Registration Authentication Extension

Figure 3.7 illustrates the format of these extensions. The *type field* is set to 32, 33, or 34 depending on if this extension is a mobile node / home agent, a mobile node / foreign agent, or a foreign agent / home agent authentication extension. The *length field* specifies the payload length of this extension, that is 4 byte plus the number of bytes of the authenticator. The *security parameter index field (SPI)* identifies the security association which specifies the security context that has to be established between the authentication peers prior to the registration exchange. In particular, the SPI selects the authentication algorithm, its mode and a secret (a shared key, or appropriate public/private key pair) used in computing the authenticator. The *authenticator* is computed over the entire Mobile IP registration message and includes all protocol fields and prior extensions up to the authenticator field itself. If a message contains more than one authentication extension, all prior authentication extensions are included in the computation of the authenticator. The default algorithm for computing the authenticator is MD5 in *prefix+suffix mode*[3].

A mobile node, that wants to register at a foreign network listens to so-called *advertisement messages* of a foreign agent. After having received such an advertisement, the registration of a mobile node is realized by running the following registration protocol:

1. The mobile node sends a registration message to the foreign agent which contains some $Flags$ concerning protocol details of Mobile IP, a requested $Lifetime$ for the registration, the home address of the mobile node $Addr_{MN}$, the address of the mobile nodes home agent $Addr_{HA}$, the care-of-address CoA the mobile node wants to register, an identifier $IdReq$ of this request, the mobile nodes network access identifier NAI_{MN}, an authentication extension to be checked by the home agent $Sig_{MN,HA}$, and optionally an authentication extension $Sig_{MN,FA}$ to be checked by the foreign agent:

$$MN \rightarrow FA\colon \quad \{RegReq, Flags, Lifetime, Addr_{MN}, Addr_{HA}, CoA, IdReq, NAI_{MN}, Sig_{MN,HA}, [Sig_{MN,FA}]\} \tag{3.21}$$

2. Upon reception of this message the foreign agent checks the mobile node / foreign agent authentication extension if present, eventually computes a foreign agent / home agent authentication

[3]This MAC construction raises some cryptographic concern [27, Example 9.66] and it is recommended to use the HMAC construction [21] instead.

extension, and sends the resulting message to the home agent:

$$FA \rightarrow HA: \quad \{RegReq, Flags, Lifetime, Addr_{MN}, Addr_{HA}, CoA, IdReq, NAI_{MN}, Sig_{MN,HA}, [Sig_{MN,FA}], [Sig_{FA,HA}]\} \tag{3.22}$$

3. The home agent checks the authenticity of the received message by re-computing the appropriate message authentication codes and comparing them to the values of the authentication extensions in the message. At least one check will be performed as the mobile node / home agent authentication extension is mandatory in the registration procedure. The home agent then creates the registration-reply message which contains the Mobile IP specific result in the $Code$ field, the $Lifetime$ of the registration, the addresses of the mobile node $Addr_{MN}$ and the home agent $Addr_{HA}$, an identifier Id_{Rep} of the reply, the network access identifier of the mobile node NAI_{MN}, the home agent / mobile node authentication extension $Sig_{HA,MN}$, and optionally the home agent / foreign agent authentication extension $Sig_{HA,FA}$. This message is send to the foreign agent:

$$HA \rightarrow FA: \quad \{RegRep, Code, Lifetime, Addr_{MN}, Addr_{HA}, IdRep, NAI_{MN}, Sig_{HA,MN}, [Sig_{HA,FA}]\} \tag{3.23}$$

4. The foreign agent checks the home agent / foreign agent authentication extension if present, eventually computes and appends the optional foreign agent / mobile node authentication extension, and sends the resulting message to the mobile node:

$$FA \rightarrow MN: \quad \{RegRep, Code, Lifetime, Addr_{MN}, Addr_{HA}, IdRep, NAI_{MN}, Sig_{HA,MN}, [Sig_{HA,FA}], [Sig_{FA,MN}]\} \tag{3.24}$$

5. Upon reception of this message the mobile node checks the included authentication extensions. If all checks are succesful and the home agent had accepted the registration request the mobile node has succesfully registered and can assume IP connectivity using his home address.

This procedure has to be repeated after expiration of the registration lifetime. At a first glance the scheme seems to be sufficient to realize all of the authentication relations mentioned above. However, it has soon been discovered that its main drawback, the missing management of keys to be shared between the mobile node and the foreign agent, as well as between the foreign agent and the home agent, represents a serious deficiency in deployment scenarios in which mobile nodes roam between multiple access networks that are operated by different providers.

While the shared secret key to be used by the foreign agent and the home agent could be established with a standard IPSec internet key exchange (IKE) [13], this is not possible for the shared key to be used by the mobile node and the foreign agent, as the MN has not yet obtained a valid IP address at the moment when it needs to establish the authentication relation to the foreign agent. Furthermore, it has to be noted that IKE is a very general purpose protocol, which offers more flexibility and requires more effort than a dedicated protocol for Mobile IP might need. This motivates integration with an authentication, authorization and accounting infrastructure.

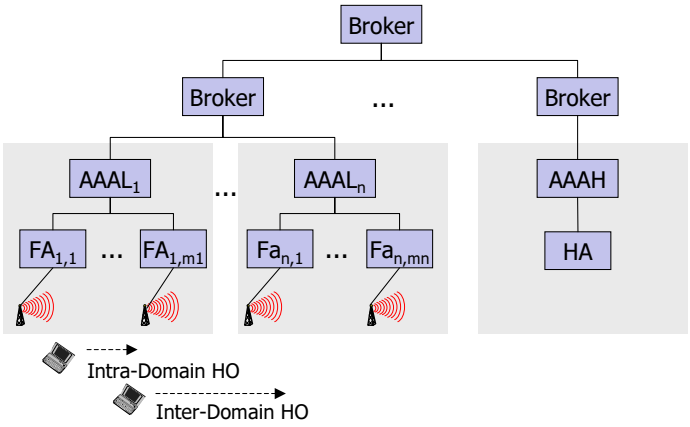


Figure 3.8: Entities involved in Integrated AAA / Mobile IP Authentication

3.4.2 Mobile IP Authentication with AAA Infrastructure

With the ubiquitous availability of dial-in services offered by *Internet Service Providers (ISP)* that emerged during the last five years, a new client demand for ubiquitous access to the service of ones ISP has grown. That means, that a client who has a service agreement with an ISP wants to make use of this agreement regardless of his current location, e.g. to access the Internet from a hotel room by using a dial-up service of a local service provider who might have an agreement with the clients ISP. The main motivation for this is reduced cost of long distance Internet communication in comparison with direct access to ones ISP over a long distance telephone line.

These kind of usage scenarios are called *roaming scenarios* and they appear in two different categories of mobile communications:

- *Nomadic communications*, in which mobility only appears between sessions, and
- *True mobile communications*, in which users may also move during active sessions.

The roaming scenario decribed above (also sometimes referred to as the *"road warrior scenario"*) falls in the category of nomadic communications and it does not necessarily have to involve a wireless network technology. In fact, as wireless Internet access using currently available wide area networks is quite slow and expensive, the original motivation for roaming operations emerged from clients accessing the Internet using the wired telephone network. However, when the IETF started working on the definition of a general authentication, authorization and accounting (AAA) infrastructure that should support roaming operation, it was soon discovered that the same infrastructure could also be used for true mobile communications, especially to support Mobile IP authentication, authorization and accounting.

Figure 3.8 shows the entities involved in a Mobile IP registration supported by an AAA infrastructure. The shaded areas mark the administrative domains to which the enclosed network entities belong. Every administrative domain contains one or more local AAA servers (AAAL) and multiple foreign agents (FA). The AAA servers of different domains may interact either directly or with the help of a

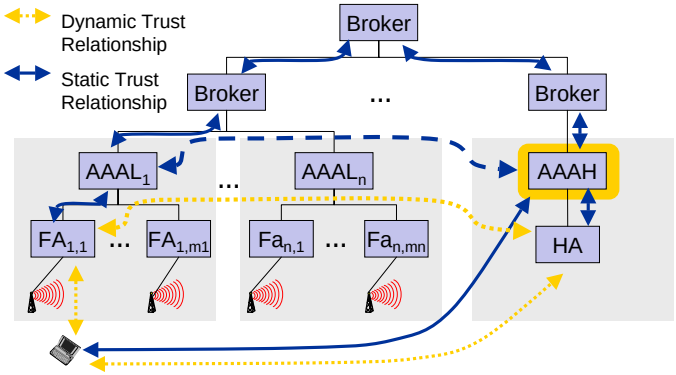


Figure 3.9: Static and Dynamic Trust Relationships in Integrated AAA / Mobile IP Authentication

network of inter-operating AAA brokers. The “home domain” of a mobile node contains one or more home AAA servers (AAAH) as well as one or more home agents (HA). As a mobile node moves it may need to change its access point necessitating a handover operation. Depending on if the old and the new foreign agent belong to the same or different administrative domains, this handover operation is called an *intra-domain handover* or an *inter-domain handover*.

The joined AAA / Mobile IP authentication procedure assumes some static trust relationships that are depicted with continuous lines in figure 3.9 and that pre-established between:

- mobile nodes and their home AAA server,
- foreign agents and their local AAA servers,
- home agents and their home AAA servers,
- AAA servers and one or more AAA brokers
- various AAA brokers, and
- eventually local and home AAA servers (segmented line) which allows to avoid the direct involvement of AAA brokers and the related performance degradation.

By making use of these static trust relationships, the AAA / Mobile IP registration procedure allows to create dynamic trust relationships which are depicted by dotted lines in figure 3.9 and are established between:

- mobile nodes and their home agent,
- mobile nodes and their foreign agent, and
- foreign agents and home agents which are currently involved in service provision for a mobile node.

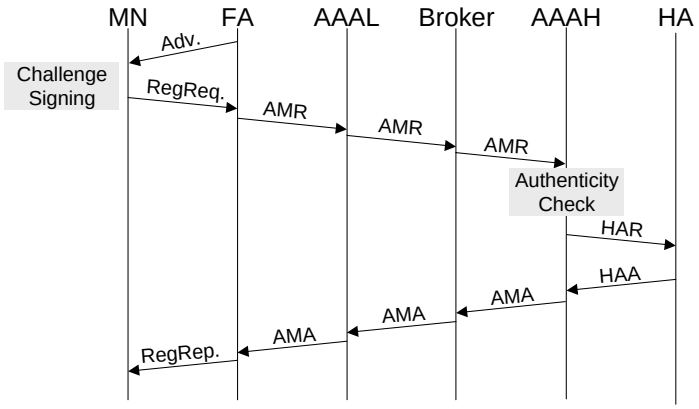


Figure 3.10: Message Flow in Integrated AAA / Mobile IP Authentication

The main motivation for realizing the trust relationship between a mobile node and his home agent dynamcally is, that this allows for dynamic assignment of a home agent to a mobile node in cases where the mobile node has no requirements regarding the home IP address with which it will register.

The joined AAA / Mobile IP registration is realized by running the following protocol (see also figure 3.10 for an overview of the message flow, the following description focusses on message contents important to the aspects of authentication and session key exchange):

1. All foreign agents periodically send out Mobile IP advertisement messages containing an NAI extension identifying themselves and a challenge-response extension which carries a random number r_{FA} freshly generated by the foreign agent:

$$FA \rightarrow MN: \quad \left(Advertisement, \ldots, NAI_{FA}, r_{FA}\right) \tag{3.25}$$

2. The mobile node stores the received NAI of the foreign agent, creates a Mobile IP registration message containing the foreign agents random number, his network access identifier and a signature that can be checked by his home AAA server, and sends this message to the foreign agent:

$$MN \rightarrow FA: \quad \left(RegReq, \ldots, r_{FA}, NAI_{MN}, Sig_{MN,AAAH}\right) \tag{3.26}$$

3. The foreign agent creates an AAA mobile registration request (AMR) message which contains the mobile nodes request message and sends it to his local AAA server:

$$FA \rightarrow AAAL: \quad \left(AMR, \ldots, RegReq, \ldots, r_{FA}, NAI_{MN}, Sig_{MN,AAAH}\right) \tag{3.27}$$

4. The local AAA server either indirectly forwards this message by the use of AAA brokers or directly sends this message to the home AAA server which can be determined by evaluating the contained network access identifier of the mobile node:

$$AAAL \rightarrow AAAH: \quad \left(AMR, \ldots, RegReq, \ldots, r_{FA}, NAI_{MN}, Sig_{MN,AAAH}\right) \tag{3.28}$$

5. The home AAA server checks the signature $Sig_{MN,AAAH}$ of the mobile node over the contained Mobile IP registration message. If this check is successful, the home AAA server may deduce, that the mobile node in fact created the registration message. However, it should be noted, that the home AAA server can not deduce anything about the freshness of the message, as it did not generate the random number challenge r_{FA} itself and, therefore, does not know when it was generated.

   The home AAA server now creates a home agent registration message (HAR) containing the mobile nodes original Mobile IP registration message, a session key $K_{MN,HA}$ for use between the mobile node and the home agent, as well as a second session key $K_{FA,HA}$ for use between the foreign and the home agent. These two session keys are encrypted with a shared secret key $K_{AAAH,HA}$. Furthermore, the home AAA server includes the session keys $K_{MN,FA}$ and $K_{MN,HA}$ to be distributed to the mobile node, encrypted with the secret key $K_{MN,AAAH}$ it shares with the mobile node[4].

   The home AAA server appends a signature to the resulting HAR message and sends it to the home agent:

$$AAAH \rightarrow HA: \quad \big(HAR, \ldots, RegReq, \ldots, NAI_{MN}, \{K_{MN,HA}, K_{FA,HA}\}_{K_{AAAH,HA}}, \{K_{MN,FA}, K_{MN,HA}\}_{K_{MN,AAAH}}, Sig_{AAAH,HA}\big) \tag{3.29}$$

6. Upon reception of this message the home agent checks the signature, registers the mobile node with the care-of-address contained in the included RegReq , and decrypts and stores the two session keys. It then creates a Mobile IP registration reply message (RegRep) which also contains the session keys as provided by the home AAA server and signature $Sig_{HA,MN}$ of the home agent. The RegRep message is inserted into a home agent answer message (HAA), and send to the home AAA server, confirming the successful registration of the mobile node:

$$HA \rightarrow AAAH: \quad \big(HAA, \ldots, (RegRep, \ldots, \{K_{MN,FA}, K_{MN,HA}\}_{K_{MN,AAAH}}, Sig_{HA,MN}), Sig_{HA,AAAH}\big) \tag{3.30}$$

7. The home AAA server creates an AAA mobile registration answer message (AMA) containing the RegRep message included in the HAA message. If the mobile node has successfully been registered at the home agent, the home AAA server furthermore includes session key material encrypted for distribution to the foreign agent. The resulting message is signed and sent to the AAA server of the visited network[5]:

$$AAAH \rightarrow AAAL: \quad \big(AMA, \ldots, r_{FA}, \{K_{MN,FA}, K_{FA,HA}\}_{K_{AAAH,AAAL}}, (RegRep, \ldots, \{K_{MN,FA}, K_{MN,HA}\}_{K_{MN,AAAH}}, Sig_{HA,MN}), Sig_{AAAH,AAAL}\big) \tag{3.31}$$

[4]The encryption of the session keys for communication to the mobile node is realized using a combination of exclusive-or and the MD5 hash function according to [35].

[5]If the message is sent to AAAL via brokers, the key $K_{AAAH,Broker}$ is used instead of $K_{AAAH,AAAL}$ and every broker performs appropriate cryptographic checks and transformations according to the hop-by-hop security model used between AAA brokers. However, for reasons of simplicity we here describe the simple case in which the home and the foreign AAA server already share a secret key.

8. The AAA server of the visited network checks the signature of the message, decrypts, stores and re-encrypts the session keys to be communicated to the foreign agent, and then sends the following message to the foreign agent:

$$AAAL \rightarrow FA: \quad \big(AMA, \ldots, r_{FA}, \{K_{MN,FA}, K_{FA,HA}\}_{K_{FA,AAAL}}, (RegRep, \ldots, \{K_{MN,FA}, K_{MN,HA}\}_{K_{MN,AAAH}}, Sig_{HA,MN}), Sig_{AAAL,FA}\big) \tag{3.32}$$

9. Upon reception of this message the foreign agent checks the contained signature and processes the AMA message. If the AMA message signals successful registration of the mobile node, the foreign agent deduces, that the mobile node had signed his random number r_{FA} correctly in the second step and can therefore be assumed to be authentic. The foreign agent decrypts and stores the contained session keys $K_{MN,FA}$ and $K_{FA,HA}$, and then forwards the RegRep message to the mobile node:

$$FA \rightarrow MN: \quad \big(RegRep, \ldots, \{K_{MN,FA}, K_{MN,HA}\}_{K_{MN,AAAH}}, Sig_{HA,MN}\big) \tag{3.33}$$

10. The mobile node first decrypts the session keys as provided by the home AAA server using the secret key it shares with the home AAA server, stores the obtained keys and then uses the key $K_{MN,HA}$ to check the signature $Sig_{HA,MN}$ that has been created by the home agent in the sixth step. If this check is positive, the mobile node has successfully been registered at the foreign agent.

If the mobile node later on needs to re-register, e.g. after expiration of the Mobile IP registration time-out, it will use the obtained session keys to sign his registration message, so that no direct involvement of the AAA infrastructure is required.

In case of a handover to another foreign agent, the mobile node will also try to perform authentication using the obtained session keys. For this it signs the new foreign agents random number challenge r_{FAnew} with the key $K_{MN,FAold}$ and indicates the identity of the old foreign agent by including the appropriate NAI extension into his registration request. In this case the authentication protocol proceeds as follows[6]:

1. The new foreign agent periodically sends out advertisement messages that contain their NAI and a random number challenge:

$$FA \rightarrow MN: \quad \big(Advertisement, \ldots, NAI_{FAnew}, r_{FAnew}\big) \tag{3.34}$$

2. The mobile node creates a Mobile IP registration request message which contains the received random number, the mobile nodes NAI, the NAI of the old foreign agent, a signature to be checked by the home agent and a signature to be checked by the new foreign agent which has been signed with the key that was previously obtained for authentication with the old foreign agent:

$$MN \rightarrow FA: \quad \big(RegReq, \ldots, r_{FAnew}, NAI_{MN}, NAI_{FAold}, Sig_{MN,HA}, Sig_{MN,FAold}\big) \tag{3.35}$$

[6]The exact procedure and message formats for this case have not yet been specified in [7], so that the description given here can just outline the basic idea that has been proposed so far. Furthermore, the description is focused on the authentication protocol and for reasons of clarity does not include all message fields

3. The foreign agent creates an AAA mobile registration request (AMR) message which contains the mobile nodes request message and sends it to his local AAA server:

$$FA \rightarrow AAAL: \quad (AMR, \ldots, RegReq, \ldots, r_{FAnew}, NAI_{MN}, NAI_{FAold}, Sig_{MN,HA}, Sig_{MN,FAold}) \tag{3.36}$$

4. The local AAA server looks up, whether it can supply the new foreign agent with the session keys $K_{MN,FAold}$ and $K_{FAold,HA}$, and, if so, updates his record and answers with a message like:

$$AAAL \rightarrow FA: \quad (AMA, \ldots, r_{FAnew}, \{K_{MN,FAold}, K_{FAold,HA}\}_{K_{FAnew,AAAL}}, Sig_{AAAL,FAnew}) \tag{3.37}$$

5. Upon reception of this message the new foreign agent decrypts the contained session keys, and using $K_{MN,FAold}$ checks the signature $Sig_{MN,FAold}$ of the mobile nodes registration request. If this check is positive it can proceed further with the normal Mobile IP registration procedure (see also step 2. in section 3.4.1).

The authentication scheme discussed so far will allow to perform intra-domain handover operations in a more efficient manner than inter-domain handovers, as the full authentication procedure involving the home AAA server can be avoided in cases where the local AAA server still holds valid session keys for the mobile node. In case of inter-domain handover operations and after expiration of the session keys lifetime, a full AAA authentication as explained before has to be performed. Furthermore, the scheme also allows to allocate the home agent in the visited domain.[7]

However, several security remarks have to be stated:

- The authentication procedure involves quite a few entities which makes security analysis difficult.
- The challenge-response verification is distributed: the foreign agent provides a random number challenge, but it can not verify the response. It has to trust a home AAA server it does only know via a chain of trust created with his local AAA server and eventually a series of AAA brokers. On the other side, the home AAA server can verify the response, but does not provide the challenge. He, therefore, may not deduce that the mobile nodes registration is fresh.
- The intermediate AAA brokers can read the session keys for the AAA server of the visited network and the foreign agent.
- The NAI extension of the mobile node is send is cleartext in the fixed network and – even worse – also over the unprotected air interface. Hence, there is no location privacy protection, not even from passive attackers eavesdropping on the air interface.

[7] In this case, the home AAA server directly answers to the foreign AAA server in Step 5 of the full dialogue and the foreign AAA server determines a home agent and performs the exchange of the HAR and HAA messages [7].

3.5 Summary

In this chapter the authentication procedures of current mobile communication standards have been presented and discussed.

The weakest security is attained by the authentication protocol of IEEE 802.11, as there is one common key for all mobile stations that are served by one base station. While IEEE 802.11 is not a mobile communications standard and does, therefore, provide only limited mobility management and handover support, this weakness should be kept in mind when "being tempted" to combine Mobile IP authentication with IEEE 802.11 authentication: it simply would not make a lot of sense, as IEEE 802.11 authentication alone is not sufficient for the above reasons and authenticating twice could only marginally (if at all) improve security.

GSM provides a scalable means for authenticating mobile users as long as they are moving in the area covered by one service provider and do not wish to perform inter-domain handovers, which are not supported. The permanent identity of a mobile station is not revealed over the air interface when this can be avoided. However, this protection can easily be circumvented by an active attacker which explicitely demands a mobile station to reveal its permanent international mobile subscriber identity. The trust model of GSM assumes trust between all operators and also assumes the signaling network to be secure – a dangerous assumption in practice given that base station transceivers are often connected via a wireless link to their base station controller. As authentication vectors are transmitted unprotected in the signaling network this enables attackers to eavesdrop on authentication vectors which could be later on used to impersonate mobile stations.

The authentication procedure of UMTS Release '99 is based on the same principles as GSM authentication. As a slight improvement the home network of a mobile station includes an authentication token into every authentication tuple which can be checked by the mobile station so that it can be assured to communicate with a network that is trusted by its home network. However, it is still possible for an active attacker to explicitely demand the permanent identification of a mobile station, as this identification is needed by the home environment to create the authentication token. As in GSM trust is assumed between all operators and in the signaling network no protection is provided for the exchanged messages.

Current approaches to authentication for Mobile IP propose an integration with AAA authentication. One of the main design goals is to realize authentication and Mobile IP registration with one single Internet traversal. As the procedures are not yet completely specified (e.g. for re-authentication in the same administrative domain) a detailed security analysis remains to be done. However, it can already be seen from the current protocol proposal that the distributed challenge-response verification (FA creates challenge, AAAH verifies the MNs response) makes security analysis more difficult. This could be avoided if the two tasks of distributing a session key and checking the timeliness of a mobile nodes response were clearly separated. Furthermore, the current proposals do not include any location privacy protection.

Chapter 4

Conclusion

This report has given a general introduction into principles and mechanisms of authentication and presented and discussed the authentication protocols of current mobile communication standards.

When designing an authentication infrastructure for mobile Internet access the following considerations should be taken into account:

- The decentralized organization of the Internet has proven to be one major advantage over classical centralized approaches, as it allows a more dynamic network evolution. However, a decentralized organization also entails drawbacks. One consequence is that the traditional trust model which assumes trust between all interworking network operators can not be upheld in the future as more and more operators, possibly including "pico-operators" of wireless local area networks, will have to interwork in order to provide at any given instance the most efficient and most economic network access to authorized users.

- As the use of wireless communications devices is to become an integral part of our daily life, increased care has to be taken of its negative impacts on privacy in order to avoid large-scale surveilance of users as well as resistance against the technology as a consequence of users feeling incomfortable with it. This requires a privacy-protecting architecture for authentication and accounting services.

4.1 Issues for Further Study

The following aspects are identified for further study:

- Security analysis: as the specification of the integrated AAA / Mobile IP procedures is not yet finished, a detailed security analysis still needs to be done. In this context two strategies are possible: to wait for the standard, analyze if it is acceptable and eventually improve the standard with some additional engineering, or to develop an own solution that meets security and performance requirements and try to influence standardization with that solution.

- Evaluation of performance implications: authentication is one task to be fulfilled during the handover of a mobile station. Therefore, it has to be taken into account when evaluating the performance issues of handover operations. In this respect, authentication during inter-domain handover operations and re-authentication occurring at intra-domain handover operations should be evaluated using standard performance analysis methods.
- Modification of protocols to improve location privacy: the current approach of the IETF is unacceptable with respect to location privacy of mobile nodes, as this is not protected at all. Therefore, further work should be done regarding the improvement of location privacy during AAA registration as well as during Mobile IP data exchange.
- Security issues of accounting tickets: as the assumption of trust between all operators becomes more and more inappropriate, more care should be taken about potential forgery of accounting tickets. One possible countermeasure could be to include periodic signing of accounting tickets by the mobile node into the signaling protocol.

4.2 Next Steps

In the further course of our work regarding authentication for mobile Internet access we will next concentrate on the performance aspects of the current IETF approach. Our first step in this direction will be the construction of a discrete event simulation model for integrated AAA / Mobile IP authentication and evaluating some selected scenarios using authentication latency, server occupation, etc. as performance metrics.

The goal of this evaluation is to get a clearer understanding of the operations critical for (re-)authentication performance. It should also provide a means to estimate appropriate dimensioning of an authentication infrastructure for Mobile IP.

Bibliography

[1] 3GPP. 3G Security: Security Architecture (Release 1999). 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects, 3GPP TS 33.102, V3.6.0, Oct 2000.

[2] B. Aboba, P. Calhoun, S. Glass, T. Hiller, P. McCann, H. Shiino, G. Zorn, G. Dommety, C. Perkins, B. Patil, D. Mitton, S. Manning, M. Beadles, P. Walsh, X. Chen, T. Ayaki, S. Sivalingham, A. Hameed, M. Munson, S. Jacobs, T. Seki, B. Lim, B. Hirschman, R. Hsu, H. Koo, M. Lipford, Y. Xu, E. Campbell, S. Baba, and E. Jaques. Criteria for Evaluating AAA Protocols for Network Access. Internet Draft, work in progress, August 2000. `http://www.ietf.org/internet-drafts/draft-ietf-aaa-na-reqts-07.txt`.

[3] T. Beth, B. Klein, and R. Yahalom. Trust Relationships in Secure Systems: A Distributed Authentication Perspective. In *Proceedings of the 1993 Symposium on Security and Privacy*, pages 150–164. IEEE Computer Society Press, May 1993.

[4] P. Bieber. A Logic of Communication in a Hostile Environment. In *Proceedings of the Computer Security Foundations Workshop III*, pages 14–22. IEEE Computer Society Press, June 1990.

[5] N. Borisov, I. Goldberg, and D. Wagner. Intercepting Mobile Communications: The Insecurity of 802.11. Draft Paper, Jan 2001. `http://www.isaac.cs.berkeley.edu/isaac/wep-draft.pdf`.

[6] M. Burrows, M. Abadi, and R. Needham. A Logic of Authentication. *ACM Transactions on Computer Systems*, 8(1):18–36, 1990.

[7] P. R. Calhoun and C. E. Perkins. Diameter Mobile IP Extensions. Internet Draft, work in progress, Mar 2001. `draft-ietf-aaa-diameter-mobileip-01.txt`.

[8] D. E. Denning and G. M. Sacco. Timestamps in Key Distribution Protocols. *Communications of the ACM*, 24(8):198–208, 1981.

[9] A. Festag, H. Karl, and G. Schäfer. Current Developments and Trends in Handover Design for All-IP wireless networks. TKN Technical Report TKN-05-00, August 2000.

[10] K. Gaardner and E. Snekkenes. Applying a Formal Analysis Technique to the CCITT X.509 Strong Two-Way Authentication Protocol. *Journal of Cryptology*, 3(2):81–98, 1991.

[11] S. Glass, T. Hiller, S. Jacobs, and C. Perkins. Mobile IP Authentication, Authorization, and Accounting Requirements. Internet RFC 2977, 2000.

[12] L. Gong, R. M. Needham, and R. Yahalom. Reasoning about Belief in Cryptographic Protocols. In *Symposium on Research in Security and Privacy*, pages 234–248. IEEE Computer Society, IEEE Computer Society Press, May 1990.

[13] D. Harkins and D. Carrel. The Internet Key Exchange (IKE). Internet RFC 2409, 1998.

[14] IEEE. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. The Institute of Electrical and Electronics Engineers (IEEE), IEEE Std 802.11-1997, 1997.

[15] ITU-T. *Draft Recommendation X.509: The Directory Authentication Framework, Version 7*, November 1987.

[16] ITU-T. *X.509: Information Technology – Open Systems Interconnection – The Directory: Authentication Framework (4)*, 1993.

[17] R. A. Kemmerer. Analyzing Encryption Protocols using Formal Description Techniques. *IEEE Journal on Selected Areas in Communications*, 7(4):488–457, 1989.

[18] V. Kessler and G Wedel. AUTOLOG – An Advanced Logic of Authentication. In *Proceedings of the Computer Security Foundations Workshop VII*, pages 90–99. IEEE Computer Society Press, 1994.

[19] J. Kohl. The Use of Encryption in Kerberos for Network Authentication. In *Proceedings of Crypto'89*. Springer, 1989.

[20] J. Kohl, B. Neuman, and T. Ts'o. The Evolution of the Kerberos Authentication Service. In F. Brazier and D. Johansen, editors, *Distributed Open Systems*. IEEE Computer Society Press, 1994.

[21] H. Krawczyk, M. Bellare, and R. Canetti. *HMAC: Keyed-Hashing for Message Authentication*, February 1997. RFC 2104.

[22] X. Lai. *On the Design and Security of Block Ciphers*. Konstanz: Hartung-Gorre-Verlag, 1992. ETH Series in Information Processing, v.1.

[23] D. Longley and S. Rigby. An Automatic Search for Security Flaws in Key Management Schemes. *Computers & Security*, 11(1):75–89, 1992.

[24] W. Mao and C. Boyd. Towards Formal Analysis of Security Protocols. In *Proceedings of the Computer Security Foundations Workshop VI*, pages 147–158. IEEE Computer Society Press, 1993.

[25] C. Meadows. Applying Formal Methods to the Analysis of a Key Management Protocol. *Journal of Computer Security*, 1(1):5–35, 1992.

[26] C. Meadows. Formal Verification of Cryptographic Protocols: A Survey. In *Advances in Cryptology – Asiacrypt '94*, number 917 in Lecture Notes in Computer Science, pages 133–150. Springer-Verlag, 1995.

[27] A. Menezes, P. Van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press LLC, 1997.

[28] M. Merrit. *Cryptographic Protocols*. Ph.D Thesis, Georgia Institute of Technology, GIT-ICS-83, February 1983.

[29] J. K. Millen, S. C. Clark, and S. B. Freedman. The Interrogator: Protocol Security Analysis. *IEEE Transactions on Software Engineering*, 13(2):274–288, 1987.

[30] L. Moser. A Logic of Knowledge and Belief for Reasoning about Computer Security. In *Proceedings of the Computer Security Foundations Workshop II*, pages 57–63. IEEE Computer Society Press, June 1989.

[31] R. Needham and M. Schroeder. Authentication Revisited. *Operating Systems Review*, 21(1), 1987.

[32] R. M. Needham and M. D. Schroeder. Using Encryption for Authentication in Large Networks of Computers. *Communications of the ACM*, 21(12):993–999, 1978.

[33] NIST (National Institute of Standards and Technology). *FIPS (Federal Information Processing Standard) Publication 46-1: Data Encryption Standard*, 1988. Aktualisiert FIPS Publication 46.

[34] D. Otway and O. Rees. Efficient and Timely Mutual Authentication. *Operating Systems Review*, 21(1), 1987.

[35] C. Perkins and P. Calhoun. AAA Registration Keys for Mobile IP. Internet Draft, work in progress, March 2001. `http://www.ietf.org/internet-drafts/draft-ietf-mobileip-aaa-key-04.txt`.

[36] G. B. Purdy, G. J. Simmons, and J. A. Studier. A Software Protection Scheme. In *Proceedings of the 1982 Symposium on Security and Privacy*, pages 99–103. IEEE Computer Society Press, April 1982.

[37] P. V. Rangan. An axiomatic Basis of Trust in Distributed Systems. In *Proceedings of the 1988 Symposium on Security and Privacy*, pages 204–211. IEEE Computer Society Press, April 1988.

[38] B. Schneier. *Applied Cryptography Second Edition: Protocols, Algorithms and Source Code in C*. John Wiley & Sons, 1996.

[39] D. P. Sidhu. Authentication Protocols for Computer Networks: I. *Computer Networks and ISDN Systems*, 11(4):297–310, 1986.

[40] G. J. Simmons. How to (Selectively) Broadcast a Secret. In *Proceedings of the 1985 Symposium on Security and Privacy*, pages 108–113. IEEE Computer Society Press, April 1985.

[41] E. Snekkenes. Exploring the BAN Approach to Protocol Analysis. In *1991 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 171–181, 1991.

[42] W. Stallings. *Cryptography and Network Security – Principles and Practice*. Prentice Hall, 1999.

[43] P. Syverson. Formal Semantics for Logics of Cryptographic Protocols. In *Proceedings of the Computer Security Foundations Workshop III*, pages 32–41. IEEE Computer Society Press, June 1990.

[44] P. Syverson. The Use of Logic in the Analysis of Cryptographic Protocols. In *1991 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 156–170, 1991.

[45] P. Syverson. Adding Time to a Logic of Authentication. In *1st ACM Conference on Computer and Communications Security*, pages 97–101, 1993.

[46] P. Syverson. On Key Distribution Protocols for Repeated Authentication. *ACM Operating System Review*, 4:24–30, October 1993.

[47] P. Syverson and P.C. van Oorschot. On Unifying Some Cryptographic Protocol Logics. In *1994 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 14–28, 1994.

[48] ETSI TC-GSM. GSM Security Aspects (GSM 02.09). European Telecommunications Standards Institute (ETSI), Recommendation GSM 02.09, Version 3.1.0, Jun 1993.

[49] ETSI TC-SMG. European digital cellular telecommunications system (Phase 2): Security related network functions (GSM 03.20). European Telecommunications Standards Institute (ETSI), ETS 300 534, Sep 1994.

[50] M.-J. Toussaint. *Verification of Cryptographic Protocols*. Ph.D Thesis, Université der Liège (Belgium), 1991.

[51] M.-J. Toussaint. Deriving the Complete Knowledge of Participants in Cryptographic Protocols. In *Advances in Cryptology — CRYPTO '91 Proceedings*, pages 24–43. Springer-Verlag, 1992.

[52] M.-J. Toussaint. Seperating the Specification and Implementation Phases in Cryptology. In *ESORICS '92 — Proceedings of the Second European Symposium on Research in Computer Security*, pages 77–101. Springer-Verlag, 1992.

[53] P. C. van Oorschot. Extending Cryptographic Logics of Belief to Key Agreement Protocols. In V. Ashby, editor, *1st ACM Conference on Computer and Communications Security*, pages 232–243, Fairfax, Virginia, November 1993. ACM Press.

[54] V. Varadharajan. Verification of Network Security Protocols. *Computers & Security*, 8(8):693–708, 1989.

[55] V. Varadharajan. Use of Formal Description Technique in the Specification of Authentication Protocols. *Computer Standards & Interfaces*, 9:203–215, 1990.

[56] T. Y. C. Woo and S. S. Lam. A Semantic Model for Authentication Protocols. In *1993 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 178–194, 1993.

Index