# Analysis and design of a SIM based authentication solution for WLAN

Technische Universität München
Fakultät für Informatik
Diplomarbeit

Aufgabenstellerin: **Prof. Anja Feldmann, Ph.D.**
Betreuer: Holger Dreger und Martin Noha
Abgabedatum: 17.09.2004

von

Robert Penz

München, 17.09.2004

Ich versichere, dass ich diese Diplomarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

Robert Penz

München, 17.09.2004

# Abstract

Within this master thesis a protocol is designed which provides a way to migrate from the at present commonly deployed web-based authentication solutions for wireless Internet access to an alternative EAP/SIM authentication solution.

As an introduction the motivation for the development of an alternative authentication scheme is given, followed by a brief overview of the basic technical background necessary for the grasp of this thesis. In the successive part the advantages and disadvantages including all fundamental problems of web-based solutions are summarized and compared to their EAP/SIM counterparts. Based on these, possible systems, which on the one hand collect most advantages and on the other hand avoid disadvantages of the currently deployed solutions, are developed.

From the proposed possible solutions the most suitable is selected for a prototypical client-server implementation. The characterization of the components of the prototype is followed by the enumeration of some problems, which occurred during practical tests. Ideas to solve these and to extend the proposed solution are shown at the end of this master thesis.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Wireless LAN has become a much discussed topic. Almost every week a new hotspot emerges [29], but it is necessary to distinguish between private and public ones. Companies which operate such public hotspots are profit oriented and therefore built facilities and systems to charge the use across their hotspots. This can be done using pre- and postpaid billing models. An example of a prepaid model is a voucher account. Postpaid billing refers to payment via credit card or payment following receipt of a telephone bill. All of them however face one or more drawbacks, which are discussed in this master thesis. Despite these problems and security concerns regarding the above methods, wireless internet service providers (WISPs) do have the infrastructure yet they are small companies [26]. They lack the money and infrastructure necessary to bill a large user base. On the other hand mobile phone operators have an enormous user base and know how to bill users on a large scale. They have become accustomed to dealing with literally millions of customers!

What would be more appropriate than to bring these two worlds together [3]?

The whole infrastructure of a mobile phone operator is centered around its cellular phones and their SIM cards. These SIM cards are used for authentication and it seems reasonable to use them in the wireless LAN world aswell. Currently the standardization of EAP/SIM is in process. The main problem with this method is that it is layer2 based and therefore needs access points to support it. This will be possible in future product generations but currently only one brand of access points supports this. So it is currently not possible to use EAP/SIM in existing hotspots. To change all access points of a large WISP is costly and an enormous burden. In addition WISPs need to make various structural modifications in order to use EAP/SIM.

The motivation behind this work is to build a solution WISPs can use for migrating to a secure authentication and communication infrastructure in the WLAN environment. This migration target is EAP/SIM. The migration to EAP/SIM is not only a subject of the Internet Engineering Task Force (IETF) but also for the European Telecommunication Standard Institute (ETSI) which is currently discussing the implementation of a part of the EAP protocol with smart cards [12].

The aim of this master thesis is therefore to develop a protocol and a prototype which can be used alongside various existing WLAN infrastructures and relying on authentication via SIM cards. The main challenge lies in utilizing the established infrastructure and if not possible without changing it to reduce the impact. There are many systems for WISPs on the market and the solution must be easy to adapt to all of them. This prerequisite does not help to make the design of the protocol between the client and the server any easier.

This master thesis has been created at the company Secartis, which deployed the first commercial EAP/SIM solution for a customer [9].

# Chapter 2

# Technical Background

To begin with, basic technologies and their origins will be explained. How they are used and interact is also shown in this section. These technologies are used in various authentication solutions described in this master thesis. The most important technologies are depicted in Figure 2.1:

- WLAN
- RADIUS
- GSM
- IEEE 802.1x and EAP
- EAP/SIM
- RADIUS server and HLR / AuC communication (SS7 will be treated briefly.)

## 2.1 WLAN

WLAN stands for Wireless Local Area Network. At the end of the 1990s the first industrial standards for wireless communication networks were defined and given names such as IEEE

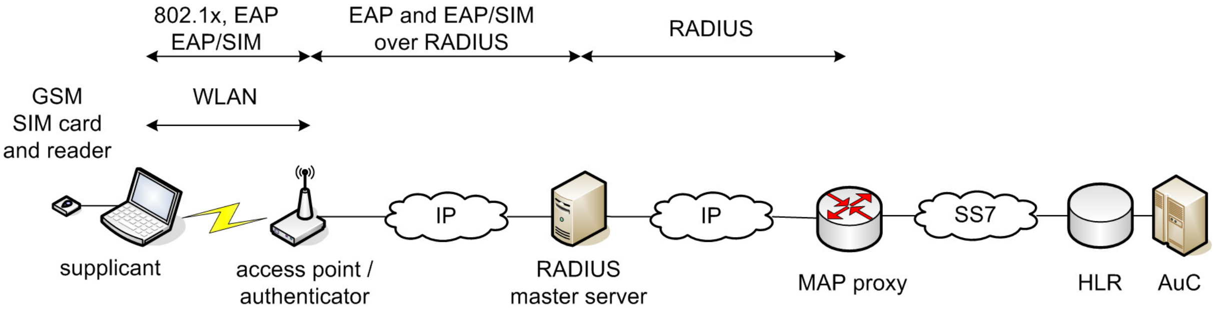


**Figure 2.1.** Basic network diagram for EAP/SIM

802.11, HomeRF (both standards with gross bandwidth of 2Mbit/s) or IEEE 802.11b (11Mbit/s) [16]. The frequency of 2,4GHz was originally reserved for industry, science and medicine use only and not for public use. But eventually the decision was made to use the ISM band (Industrial, Scientific, Medical) also for wireless communication techniques like Bluetooth and WLAN. Nowadays IEEE 802.11b is the most common standard in use [14].

Many amendments to the standard were defined to increase the bandwidth. IEEE 802.11g is in a special position here as it is downwardly compatible with IEEE 802.11. Also commonly known is the IEEE 802.11a standard which works in the 5GHz band. Both standards reach a gross bandwidth of 54Mbit/s. Table 2.1 shows all important WLAN related standards and their full names [37].

## 2.2 RADIUS

The RADIUS protocol has been originally designed to be able to validate dial-up users [28]. RADIUS stands for Remote Authentication Dial In User Service and was standardized as RFC2058 in 1997. Since it was replaced by the RFC2138, which has been replaced by the RFC2865 and its extension RFCs [8][4][27]. RADIUS supports the well-known three A's:

- authentication
- authorization
- accounting

| Standard | Full name |
|---|---|
| IEEE 802.11 | Working Group for Wireless LANs [17][2] |
| IEEE 802.11a | High Data Rate Extension (6/12/24Mbit/s, opt. 9/18/36/54 Mbit/s) [18][1] |
| IEEE 802.11b | High Data Rate Extension (5.5/11Mbit/s) [19] |
| IEEE 802.11b-cor1 | Corrigendum to the MIB [20] |
| IEEE 802.11e | MAC Enhancements for Quality of Service |
| IEEE 802.11f | Reccommended Practice for Inter Access Point Protocol [23] |
| IEEE 802.11g | Standard for Higher Rate (>20 Mbps) Extensions in the 2.4 GHz Band [24] |
| IEEE 802.11h | SMa - Spectrum Managed 802.11a [25] |
| IEEE 802.11i | Authentication and Security |
| IEEE 802.1x | Port Based Network Access Control [22] |
| IEEE 802.15 | Working Group for Wireless Personal Area Networks (WPANs) |
| IEEE 802.16 | Working Group on Broadband Wireless Access Standards (Standard for Wireless Metropolitan Area Networks) |

**Table 2.1.** Important WLAN-related standards

The RADIUS architecture supports dial-in servers, called network access servers (NAS), which are deployable in any telecommunication company's backbone and can be accessed by the customer (i.e. telecommunication company) without the need to make any specific changes. Therefore it enables a centralized user management [11].

Figure 2.2 shows a typical dial-up system. When using such a system, it is necessary to take the following steps to establish a connection:

- The user connects to the NAS via modem or ISDN.
- Having successfully established a line connection between the modem and the NAS, the client sends his authentication credentials. This is done via a PPP internal authentication protocol such as PAP or CHAP.
- These credentials are transferred via the RADIUS protocol to the RADIUS server, which checks the validity of the user's data. The check on the RADIUS server is possible via a user database, which for example can be stored on a SQL server. The result, accept or reject, is then sent back to the NAS, which depending on the result grants or denies access.
- In addition it is possible to send further information with the RADIUS packet, for example which services the user is allowed to use.

For assessing the benefits of the RADIUS protocol the following should be noted:

- As seen in Figure 2.2 the NAS is mainly a protocol conversion device from PPP internal authentication to RADIUS.
- In common setups the NAS does not directly communicate with the actual authenticating server. It is common to use a RADIUS proxy which forwards the packets to the appropriate RADIUS server.
- In order to support roaming, which requires a cascade of RADIUS server, the user name contains a postfix called "realm". This realm is separated by an @ from the user name. The realm helps the RADIUS server to forward the request for authentication to the correct database or RADIUS server [5].
- RADIUS is not only used in dial-in or WLAN authentication solutions, but also for authenticating users on VPN gateways, firewall appliances or even web servers.

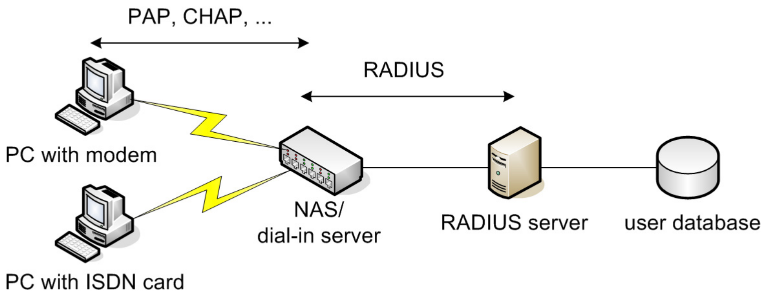


**Figure 2.2.** Components of dial-in access

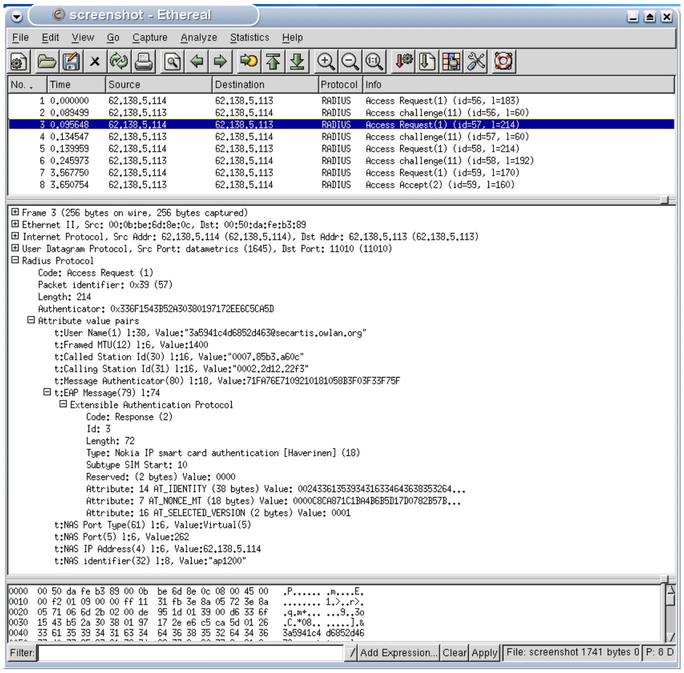
To understand the RADIUS packet itself it is helpful to cast a glance at a sniffed (Figure 2.3) and decoded packet.

- The packet is send via UDP. It is retransmitted if no acknowledge packet is received some time frame.
- The data is stored as a list of attributes and value pairs, called TLV (tag length value).
- The packets are binary encoded, using the full ASCII range.
- To ensure at least some security the user password is encrypted with a shared secret which is configured in both configurations. The encryption is done by MD5-hashing the shared secret, followed by merging the MD5 output and the user password using XOR.

## 2.3 GSM

The GSM network (Global System for Mobile communication) is the leading mobile phone standard worldwide. This section summarizes it to a degree necessary in order to understand this thesis. Further information about this protocol can be found in [36].

As within the conventional telephone network the subscriber identification is related to his fixed telephone number. In the conventional telephone system the phone number is bound to a telephone jack, in the GSM system it is bound to the Subscriber Identity Module (SIM). For confidentiality reasons GSM distinguishes between the call number MSISDN (Mobile Subscriber ISDN [Integrated Service Digital Network]) and the subscriber identity IMSI (International Mobile Subscriber Identity)[15]. Only the MSISDN needs to be known to the public. The unique



```
No. .  Time       Source         Destination    Protocol  Info
     1 0.000000   62.138.5.114   62.138.5.113   RADIUS    Access Request(1) (id=56, l=183)
     2 0.089499   62.138.5.113   62.138.5.114   RADIUS    Access challenge(11) (id=56, l=60)
     3 0.095648   62.138.5.114   62.138.5.113   RADIUS    Access Request(1) (id=57, l=214)
     4 0.134547   62.138.5.113   62.138.5.114   RADIUS    Access challenge(11) (id=57, l=60)
     5 0.139959   62.138.5.114   62.138.5.113   RADIUS    Access Request(1) (id=58, l=214)
     6 0.245973   62.138.5.113   62.138.5.114   RADIUS    Access challenge(11) (id=58, l=192)
     7 3.567750   62.138.5.114   62.138.5.113   RADIUS    Access Request(1) (id=59, l=170)
     8 3.650754   62.138.5.113   62.138.5.114   RADIUS    Access Accept(2) (id=59, l=160)
```

```
⊞ Frame 3 (256 bytes on wire, 256 bytes captured)
⊞ Ethernet II, Src: 00:0b:be:6d:8e:0c, Dst: 00:50:da:fe:b3:89
⊞ Internet Protocol, Src Addr: 62.138.5.114 (62.138.5.114), Dst Addr: 62.138.5.113 (62.138.5.113)
⊞ User Datagram Protocol, Src Port: datametrics (1645), Dst Port: 11010 (11010)
⊟ Radius Protocol
    Code: Access Request (1)
    Packet identifier: 0x39 (57)
    Length: 214
    Authenticator: 0x336F1543B52A30380197172EE6C5CA5D
  ⊟ Attribute value pairs
      t:User Name(1) l:38, Value:"3a5941c4d6852d463@secartis.owlan.org"
      t:Framed MTU(12) l:6, Value:1400
      t:Called Station Id(30) l:16, Value:"0007.85b3.a60c"
      t:Calling Station Id(31) l:16, Value:"0002.2d12.22f3"
      t:Message Authenticator(80) l:18, Value:71FA76E7109210181058B3F03F33F75F
    ⊟ t:EAP Message(79) l:74
      ⊟ Extensible Authentication Protocol
          Code: Response (2)
          Id: 3
          Length: 72
          Type: Nokia IP smart card authentication [Haverinen] (18)
          Subtype SIM Start: 10
          Reserved: (2 bytes) Value: 0000
          Attribute: 14 AT_IDENTITY (38 bytes) Value: 00243361353934316334643638353264...
          Attribute: 7 AT_NONCE_MT (18 bytes) Value: 0000C8CA871C1BA4B6B5D17D0782B57B...
          Attribute: 16 AT_SELECTED_VERSION (2 bytes) Value: 0001
      t:NAS Port Type(61) l:6, Value:Virtual(5)
      t:NAS Port(5) l:6, Value:262
      t:NAS IP Address(4) l:6, Value:62.138.5.114
      t:NAS identifier(32) l:8, Value:"ap1200"
```

```
0000  00 50 da fe b3 89 00 0b  be 6d 8e 0c 08 00 45 00   .P...... .m....E.
0010  00 f2 01 09 00 00 ff 11  31 fb 3e 8a 05 72 3e 8a   ........ 1.>..r>.
0020  05 71 06 6d 2b 02 00 de  95 1d 01 39 00 d6 33 6f   .q.m+... ...9..3o
0030  15 43 b5 2a 30 38 01 97  17 2e e6 c5 ca 5d 01 26   .C.*08.. .....].&
0040  33 61 35 39 34 31 63 34  64 36 38 35 32 64 34 36   3a5941c4 d6852d46
```

Figure 2.3. Ethereal snapshot showing a RADIUS packet

mapping MSISDN $\Longleftrightarrow$ IMSI is done in the HLR (Home Location Register). Thus the IMSI, which is used for identification, is normally not known to anyone outside the network operation personnel of the GSM operator. The IMSI is stored in the SIM, to identify the SIM to the network.

The IMSI and the assigned key Ki are integral parts of the mechanism to ensure the subscriber identity and confidentiality. The key Ki is stored safely way on the SIM card and is never transmitted through the air interface. On the mobile phone operator side the Ki is stored at the AuC (Authentication Center) of the home network operator.

GSM authentication uses a "challenge and response" method, the A3/A8 authentication algorithm. It runs on the SIM card and gets a 128bit random number (RAND) as challenge. The SIM calculates based on the RAND and the Ki a 32bit response SRES and a 64bit key Kc with an operator specific algorithm. The Kc key is used to encrypt the air interface of GSM [34]. A GSM authentication triplet is a tuple containing the three GSM authentication credentials RAND, Kc and SRES.

## 2.4 IEEE 802.1x and EAP

The use of IEEE 802.1x offers an effective framework for authenticating and controlling user traffic into a protected network. One of its nice features is that it dynamically varies encryption keys [22]. 802.1x uses the Extensible Authentication Protocol (EAP) originally specified for dial-up access. Here it uses Ethernet LANs (EAPOL) or wireless (EAPOW) as access technique. For details specifically on EAP, see IETF's RFC3748 [6]. EAP supports extensions for multiple authentication methods, such as SIM cards, public key certificates and password hashes but also methods for token cards, Kerberos, one-time passwords, certificates, and public key authentication are defined. Table 2.2 shows the most widely-deployed EAP methods.

In the beginning of an 802.1x communication the supplicant is unauthenticated (i.e., client device). The authentication is started when the supplicant attempts to connect to an authenticator (i.e., 802.11 access point). The access point responds by enabling a port which only allows EAP packets from the client to an authentication server located on the wired side of the access point. The access point blocks all other traffic, such as IP, ICMP and DHCP packets, until the access point can verify the client's identity using an authentication server (e.g., RADIUS). Once authenticated, the access point opens the client's port for other types of traffic.

To better understand how 802.1x and EAP operate Figure 2.4 illustrates the steps between the various 802.1x elements.

1. The start of an authentication depends on the configuration of the authenticator. After a successful association the authenticator sends an EAP identity request without supplicant interaction or the supplicant explicitly requests the EAP identity request via an EAPOL packet.

2. The supplicant answers that request with an EAP identity response. The authenticator forwards this packet to the authentication server which uses one of the authentication algorithm from Table 2.2.

3. Depending on the chosen authentication method challenge packets are exchanged between the supplicant and the authentication server via the authenticator.
4. The authentication server sends either a acceptance or reject message to the authenticator.
5. The authenticator sends an EAP success or reject packet to the supplicant. If the authentication server accepts the supplicant, the authenticator sets the port to authorized state and forwards additional traffic.

## 2.5 EAP/SIM

EAP/SIM specifies a mechanism for authenticating a supplicant to a network combined with a session key agreement via a GSM SIM card. The authentication is mutual. EAP/SIM also proposes some enhancements to the GSM authentication procedure. EAP/SIM currently lies with the Internet Engineering Task Force (IETF) for purpose of standardization as RFC [13]. During the time of development two companies in particular made the largest contribution, namely Nokia and Cisco which also employ the two main authors.

As EAP/SIM is an EAP method, the EAP/SIM frames are packed into RADIUS packets at the access point. As authenticator the access point only needs to support IEEE 802.1x and EAP [30] [7]. Currently only a few RADIUS servers - all commercial - provide a plugin which is able to handle EAP/SIM. The RADIUS server uses triplets to challenge the client. These triplets are provided by an HLR (Home Location Register) or AuC (Authentication Center).

| **Method** | **Server Authentification** | **Supplicant Authentification** | **Dynamic Key Delivery** | **Security Risks** |
|---|---|---|---|---|
| **EAP/MD5** | None | Password hash | No | Identity exposed, Dictionary attack, Man-in-the-Middle (MitM) attack, Session hijacking |
| **LEAP** | Password hash | Password hash | Yes | Identity exposed, Dictionary attack |
| **EAP/TLS** | Public Key (Certificate) | Public Key (Certificate or Smart Card) | Yes | Identity exposed |
| **EAP/TTLS** | Public Key (Certificate) | CHAP, PAP, MS-CHAP(v2), EAP | Yes | MitM attack |
| **PEAP** | Public Key (Certificate) | Any, EAP like EAP-MS-CHAPv2 or Public Key | Yes | MitM attack |
| **EAP/SIM** | yes (more in section 2.5) | GSM SIM card | Yes | Identity exposed |
| **EAP/AKA** | yes (same as EAP/SIM) | UMTS SIM card | Yes | Identity exposed |

**Table 2.2.** Widely-deployed EAP authentication methods.

### 2.5.1 Origin and Improvements compared to GSM

The EAP/SIM protocol improves the GSM authentication and key exchange, in that it combines multiple authentication triplets. This enables the generation of stronger authentication answers and session keys.

Among the weaknesses of GSM authentication are the lack of mutual authentication and the fact that 64bit encryption keys are not sufficient for data networks. 128bits are considered as the minimum key length in data networks. To achieve this using multiple RANDs (random number) pose the challenges to obtain multiple Kc keys, which are then combined to a strong key material. The Kc keys are therefore not used directly at EAP/SIM but are utilized to derive stronger keys. The GSM authentication is also improved by a Message Authentication Code (MAC) to ensure the integrity of the authentication. With EAP/SIM the client issues a random number NONCE_MT to the network, in order to contribute to the key derivation. This prevents replay of EAP/SIM requests from previous exchanges.

### 2.5.2 Authentication and encryption differences between EAP/SIM and GSM

There are four main differences:

- In the case of GSM only the client is authenticated and there is no way for the client to check if he is connected to a legal GSM provider. EAP/SIM provides a mutual authentication to prevent a man-in-the-middle attack as is possible with GSM.

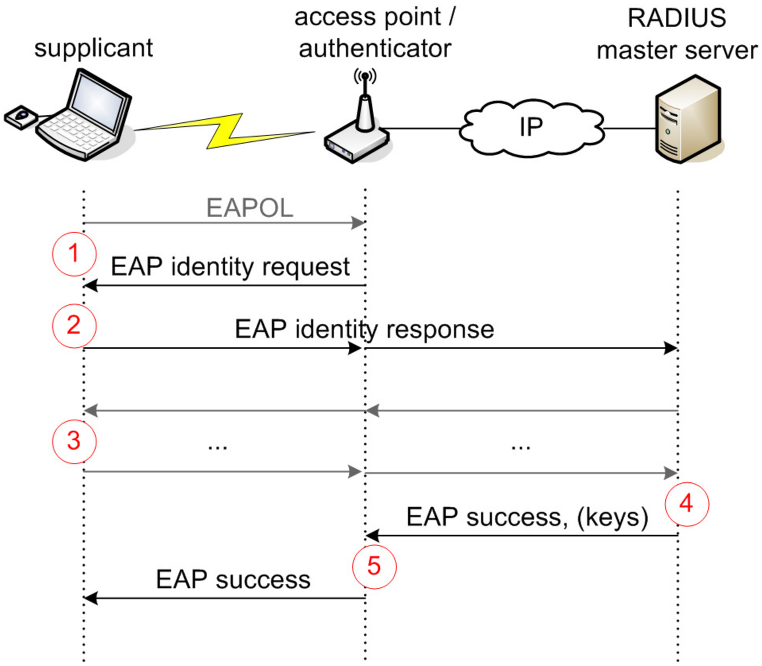


**Figure 2.4.** IEEE 802.1x authentication using EAP and EAPOL

- In the GSM standard the key which is used to encrypt the air interface is only 64bit long. In EAP/SIM it can be up to 128bit long [31]. The words "up to" are used because it depends on operator specific parameters including authentication algorithms, the strength of the Ki key, and the quality of the RAND challenges. For example some SIM cards generate Kc keys with 10 bits set to zero. Such restrictions can prevent the concatenation technique from yielding strong session keys.
- To ensure the integrity of an EAP/SIM authentication the core authentication packets are hashed with HMAC-SHA1-128 as defined in the RFC 2104. (The HMAC-SHA1-128 value is obtained from the 20byte HMAC-SHA1 value by truncating the output to 16bytes.)
- The AT_IV and AT_ENCR_DATA attributes can be used to transmit encrypted information between the EAP/SIM client and server. Transmitting the TMSI or a re-authentication identity are 2 examples of tags transmitted encrypted.

All four differences render the EAP/SIM authentication superior to GSM with regards to security.

### 2.5.3 Authentication methods and identity types

This section discusses the two methods of authentication and the various identity types in EAP/SIM. The content of this section is necessary to understand the EAP/SIM protocol, which is important to comprehend some challenges within this thesis. Three identity types are distinguishable:

**permanent identity (IMSI):** The permanent identity contains typically the IMSI of the SIM card and a Realm. It is only used during a full authentication.

**pseudonym identity (TMSI):** The pseudonym identity is transfered to the user during a full authentication within the encrypted part of an EAP/SIM packet. The identity may be used for a full authentication instead of the permanent identity. This protects the IMSI as it reduces the number of times it is transited in clear text through the air.

**re-authentication identity:** During a full authentication, a re-authentication identity will be issused to the client in the encrypted part. The client may use this identity to carry out a

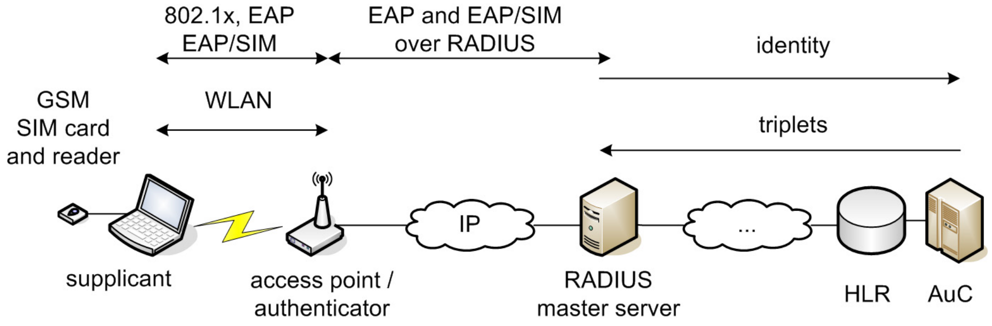


**Figure 2.5.** Flow of EAP/SIM packets

re-authentication without accessing the SIM card.

Some client implementation send "dummy IDs" to the RADIUS server within the EAP identity response packet. These pseudonyms are not known to the RADIUS server and the server therefore requests a real ID via an EAP/SIM packet. Dummy IDs are not defined in the EAP/SIM specification but are also not prohibited.

### a) Full authentication

The full authentication is the base of the EAP/SIM protocol, with which it is possible to definite identify a SIM card. The following steps take place during a full authentication, see Figure 2.6 and rely on triplets from the HLR/AuC.

1. The authentication begins with an EAP identity request packet from the authenticator.
2. The client transmits the permanent identity or the permanent pseudonym, from which the RADIUS server derives the permanent identity.
3. As the supplicant may use a permanent pseudonym unknown to the RADIUS server, the RADIUS server may request different identities until it has identified the supplicant.

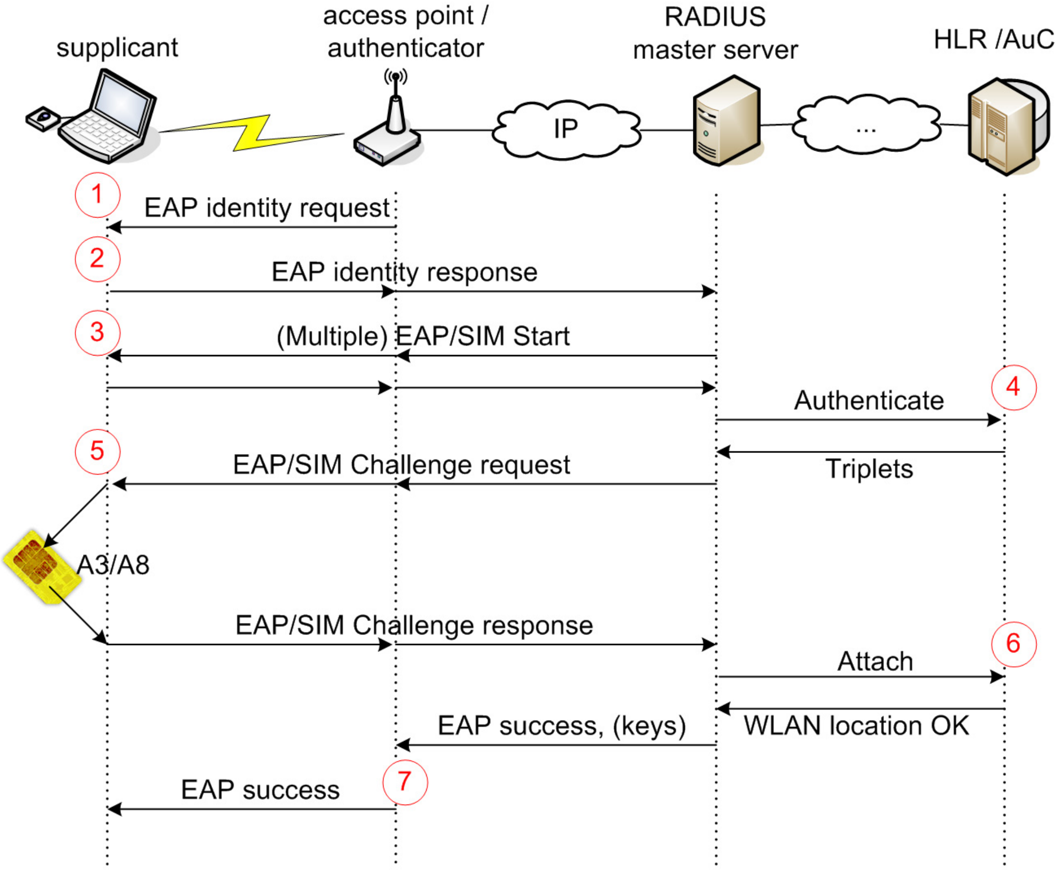


**Figure 2.6.** Sequence of steps for full authentication

Within the EAP/SIM start packets, the supplicant and the authentication server also agree to a version of EAP/SIM.

4. For this identity the RADIUS server requests the necessary triplets from the HLR or AuC. These triplets contain the challenge and the result of the A3/A8 algorithms.
5. The RADIUS server sends the challenge to the client and then compares the answer with the result it received from the HLR or AuC. The supplicant calculates the answer within the SIM card.
6. After the successful authentication an authorization procedure is carried out.
7. This is followed by the RADIUS server sending the key material, for encrypting the air interface to the access point. The client has already derived the keys and so only receives the success message without the keys.

### b) Re-authentication

As during re-authentication the client does not access the SIM card, which takes some seconds, it is faster than full authentication. The time necessary for authentication is also decreased by the fact that fewer packets are exchanged. A re-authentication is only possible after a full authentication, because the client requires the temporary pseudonym, which is obtained during full authentication. In principle this full authentication does not have to happen in the same session. But since it can be a security problem it should be prevented by the RADIUS server. The RADIUS server should not accept a re-authentication at session start.

Re-authentication provides the RADIUS server with the possibility to check if a client is still alive. During re-authentication a new air interface encryption key is calculated. It is also possible for an operator to logout an active user with a forced re-authentication. This is done by rejecting the re-authentication request. Figure 2.7 illustrates the steps during re-authentication.

1. The re-authentication begins with the access point sending an EAP Identity Request to the client.
2. The client returns a re-authentication identity to the access point, which forwards the packet to the RADIUS server.
3. The exchange of EAP/SIM start packets is optional in re-authentication as there is no need for a version negotiation. The server can explicitly request a re-authentication identity if the client did not send a valid one during step 2.
4. The RADIUS server, if it detects a valid re-authentication identity, sends a re-authentication request instead of an EAP/SIM Challenge request. The RADIUS server and the client calculate new session keys for encrypting the connection to the access point, using the master key from full authentication.
5. The RADIUS server sends the session keys to the access point together with the success message.

As the client does not need to access the SIM card, the RADIUS also does not need to access the HLR. This reduces the load on the HLR and AuC.

### c) Obligatory and optional protocol parts

The EAP/SIM draft includes two authentication methods mentioned above. Together with the three different identities three core elements of the EAP/SIM protocol are identifiable:

- full authentication with permanent identity (obligatory for every server and client)
- full authentication with a pseudonym identity (optional for client and server)
- re-authentication with a re-authentication identity (optional for client and server)

## 2.5.4 Current state of the EAP/SIM protocol

The standardization of EAP/SIM is proceeding in two steps. The first one is the EAP/SIM protocol version 1 and the second one includes future versions of the protocol.

### a) EAP/SIM protocol version 1

The aim of development was the standardization of version 1. The current draft, at the time of writing this document, is draft number 13, which was submitted to the IETF on the 5.4.2004

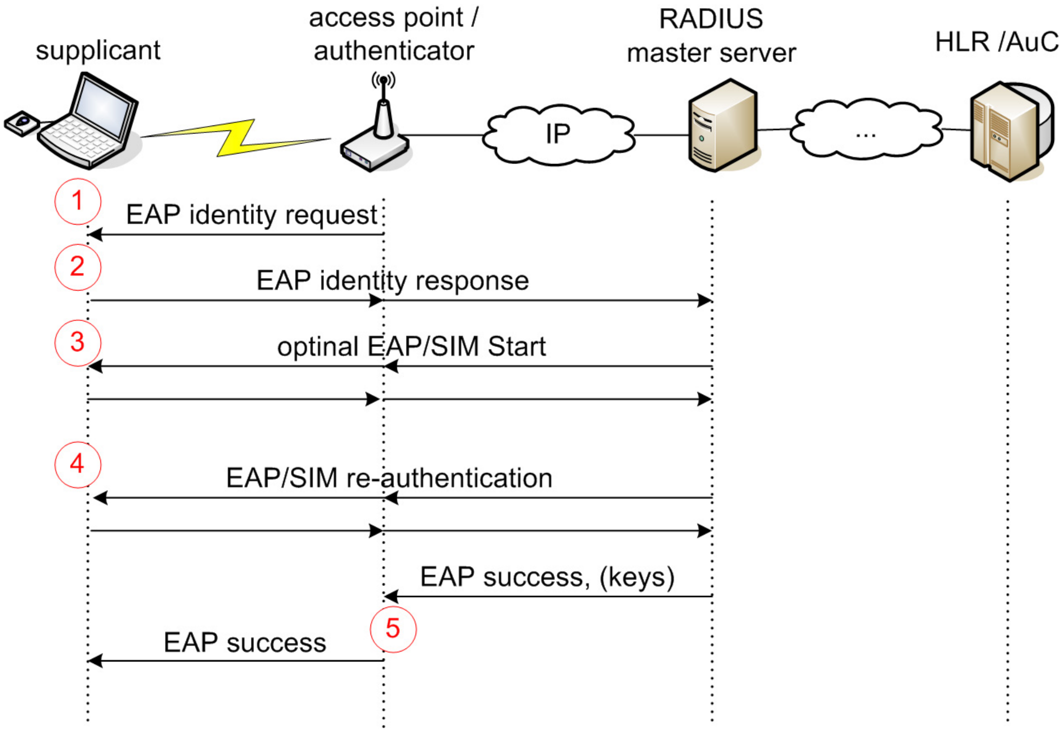


**Figure 2.7.** Sequence of steps for re-authentication

for standardization. If it passes its review, it will be released as an RFC. Note that starting with draft 08 only minor changes have been made to the protocol. In particular the cryptographic elements hardly changed at all.

There are also no fundamental changes expected during the IETF review, for example a proposed change to the encryption at the end of last year was rejected to preserve downwards compatibility. Currently only two things can force changes:

- While the standard arises in the IP world, EAP/SIM also touches GSM standards. Therefore a special G3P taskforce made some proposals. The results of both work groups differ in some elements. For example they propose different realms.
- Some details of the security and functional aspects may change again. Afterall these are the parts with the largest changes between 11, 12 and 13. For example the client is now allowed to reject an authentication if the server does not send new triplets or if a triplet is used more than once through authentication.

### b) Future versions of the protocol

Currently no plans exist to develop a version 2 of EAP/SIM. Still the possibility of future versions was considered during the design of version 1. It includes a version negotiation using the EAP/SIM start packets. It is also possible to use more than one version in parallel during migrating the clients. It is only necessary to change the RADIUS server to support both versions. The the access point only implements IEEE 802.1x and EAP.

## 2.6 RADIUS server and HLR / AuC communication

This subsection explains how the RADIUS server obtains the triplets from the HLR/AuC and describes a typical productive environment setup.

The RADIUS server with the EAP/SIM plugin requests triplets from the HLR/AuC, here the problem is often that the HLR/AuC is in an SS7 network. This is a special non IP network which is used by GSM providers for their internal communication. A conversion device needs to be deployed between the IP and SS7 based network. The conversion device has to understand RADIUS on the one side and SS7 MAP on the other. The RADIUS server with EAP/SIM plugin then sends the triplets request via the RADIUS protocol to the conversion device which then sends the request as SS7 MAP SendAuthInfo request to the HLR/AUC. Once the conversion device receives the SS7 MAP SendAuthInfo response it generates the corresponding response RADIUS packet. Such devices, RADIUS server with EAP/SIM plugin and an interface into the SS7 world, are commercial available.

# Chapter 3

# Existing WISP systems implementations

This chapter covers currently public available solutions for realizing WISP systems including those using EAP/SIM. For each solution we consider the user perspective, the backend setup and then finish with an evaluation.

## 3.1 Web-based solutions

Web-based solutions are discussed first as they represent the largest group of authentication solutions currently on the market. We start by taking a look at the various types of web-based authentications. Next the customers viewpoint towards them is explained. Then a discussion about the necessary infrastructure and an evaluation of the solution follows.

### 3.1.1 Use case

A user visits a hotspot and wants to connect to the Internet. First he needs to find the wireless network that he wants to connect to. If more than one network is available he needs to choose from a list. After he chooses the right network he will get an IP address and other configuration information via DHCP. He now simply starts the browser, and as most users access a start page. The browser tries to resolve the DNS name of that site and tries to connect to the given IP address. This connection attempt is intercepted and forwarded by the PAG (Public Access Gateway) to a backend server of the WISP. While this server could just return a page this would poison the browser cache and it would show the WISP site under a foreign URL in the browser window. To solve these problems the WISP server carries out an HTTP redirect to its real site.

On this site, often called portal, the user has the ability to choose from various authentication methods. In this scenario he may only need to supply a user name and password, as he uses an account which is billed monthly. After entering the data, a page is shown which confirms his login. On some WISP systems he will now be redirected to the site he tried to access before he was redirected to the portal site.

### 3.1.2 Diverse Web-based methods

The Web-based solution includes a large group of various authentication and billing methods. Among them are:

- User name and password: The user is billed once a month.
- Voucher account, is an account that is created in advance and can be used by users who purchase the right to access the Internet e.g. for 2 or 24 hours
- Pre-payed accounts, are accounts where the user has the right to use the service until his credits are consumed.
- Billing via mobile phone bill, the user enters his mobile phone number and then receives an access code via SMS, which he enters on the portal site.
- Direct credit card billing: The WISP system creates a hidden account after the user has provided his credit card data. The user is not aware of the created account.
- A coin based pay model where also a hidden account is created is currently in the test phase.

While there are other Web-based solutions, these are the most commonly used ones. The coin and credit card methods are included in this list since even, if invisible to the user, some kind of code has to be used to create their hidden accounts.

### 3.1.3 Infrastructure

Two different designs of the infrastructure of a WISP are widely used, central and decentral, see Figure 3.1.

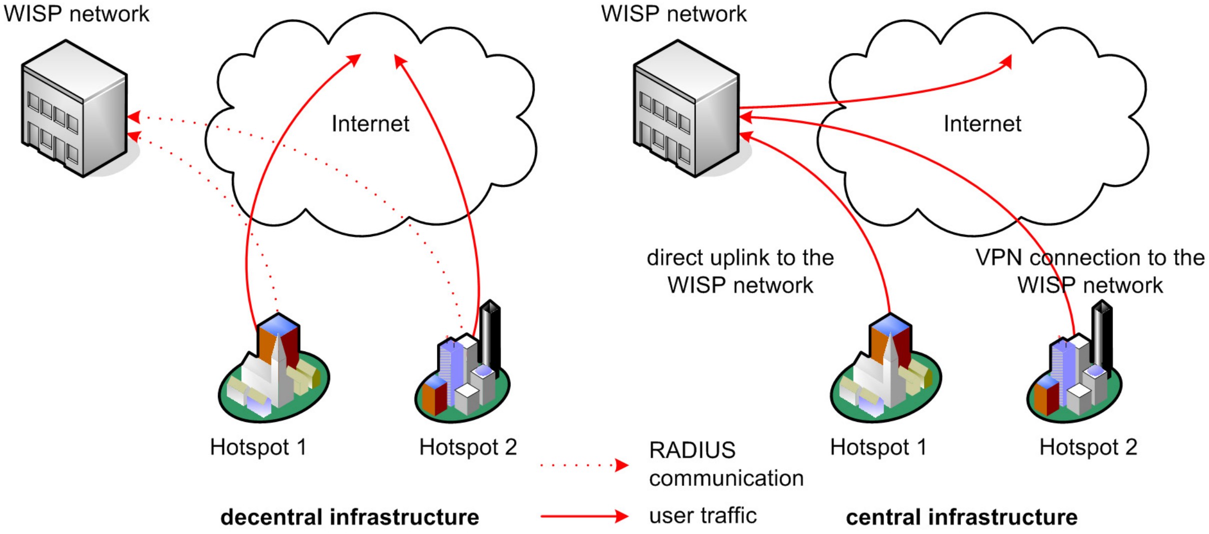


**Figure 3.1.** Central and Decentral WISP infrastructure

In the central infrastructure the traffic into the Internet is routed through the WISP network while in the decentral one the traffic from the hotspots is routed directly into the Internet. Using the central infrastructure the hotspots connect directly to the WISP or create VPNs from the hotspot to the network of the WISP. The central solution is often also refered to as Cisco solution as Cisco is the only big WISP infrastructure market player, using this solution. The decentral method is used by almost every other WISP system provider, such as Garderos and Axiros wireless creation.

Both solutions look simular to the customer, as long as he does not run a traceroute to inspect the path his packets take to the Internet. To illustrate the differences in infrastructure between the two setups, both will be discussed in this section. The advantages and disadvantages will not be analysed as this would out of the scope of this thesis.

## a) Central solution

The hotspots are connected directly with the WISP network or via VPNs, as shown in Figure 3.2. At the hotspot site only an access point and a router is needed. These may be implemented in one physical device. As the access point is configured as bridge the client can retrieve his IP address from the DHCP server on the router. The hotspot router is just a router, optional with VPN client support, and does not block non-authenticated users. This blocking is done at a server within the WISP network center.

An example for such a server is the Service Selection Gateway (SSG) from Cisco. The SSG blocks every non port 80 packet and redirects port 80 traffic to a WISP web server. The web server queries the user database via the RADIUS protocol, which is intercepted by the SSG

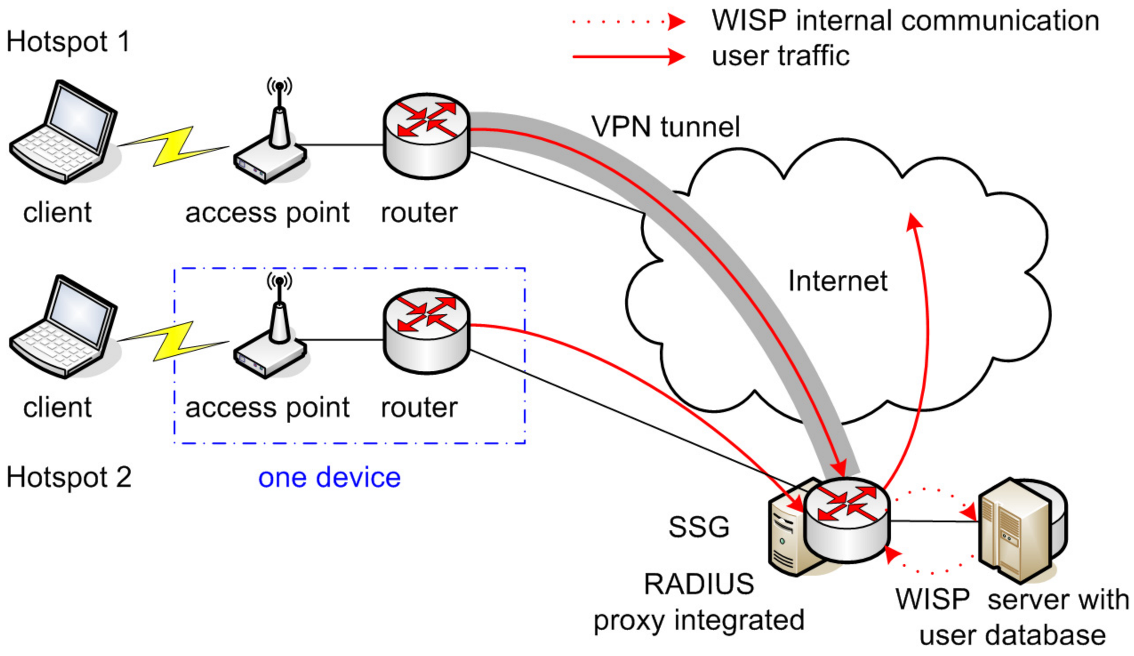


**Figure 3.2.** Principle central structure of a WLAN provider

RADIUS proxy. This is necessary as the SSG needs to know when a user is authenticated. After the RADIUS accept packet has been intercepted by the SSG, the client connection to the Internet is activated.

### b) Decentral solution

The hotspots are connected to the Internet and have no user traffic VPN to the WISP network. At the hotspot site only an access point and a router is needed. These may be implemented in one physical device. As the access point is configured as bridge the client can retrieve his IP address from the DHCP server on the router.

Before the client is authenticated the PAG (Public Access Gateway) blocks every non port 80 packet and redirects port 80 traffic to a WISP server. This internal communication may be made via VPN or directly. The client authenticates using a portal of the WISP. After authentication the WISP server instructs the PAG to forward packets from the client with this IP and MAC address to the Internet. Figure 3.3 illustrates the setup in detail.

## 3.1.4 Evaluation

Web-based methods have following advantages and disadvantages.

### a) Advantages

- The advantage of the web-based methods lies within their spontaneous character: they allow the use of voucher accounts or credit cards.
- Another advantage is the use of standard software. No specific client programs are needed which ensures operating system independency.

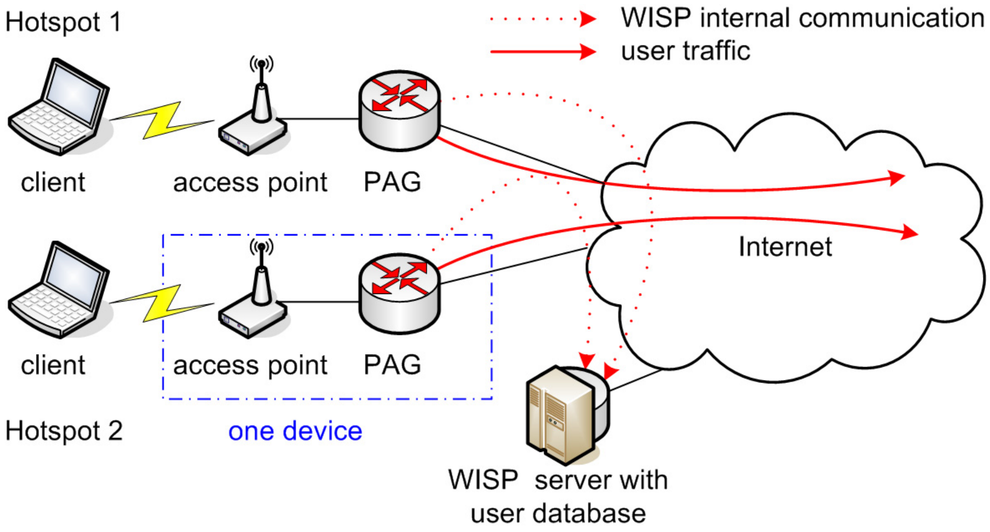


**Figure 3.3.** Principle decentral structure of a WLAN provider

#### b) Disadvantages

Among the disadvantages are security problems:

- A Trojan horse on the client can spy all necessary informations.
- The air connection is not encrypted.
- Using simple IP address and MAC address spoofing it is possible to connect to the Internet. As many users do not logout, it is also possible to take over the connection of another user that has already left.

Further the following problems arise in roaming:

- While it is theoretically possible, each provider uses its own portal and therefore looks different to the user. While this is a security problem, the user also needs to get used to the portal pages of every roaming partner he uses. This means that roaming is possible only on a small scale. Do not try Europe wide roaming using this approach.
- WLAN is an interesting business case for mobile phone operators, yet their infrastructure is based on SIM cards. If they stuck with this solution the mobile phone operators would need to send every user a letter with user name and password the user would have to memorize. Furthermore the operator would need to switch from an IMSI based billing to something different, increasing the costs. Therefore a wide-spread adaption to this solution by mobile phone operators is not very realistic.

In summary web-based solutions are ideal for spontaneous users but lack security and interoperability with existing user classes such as GSM users. For longer customer retention, other methods such as SIM based solutions are preferable.

## 3.2 EAP/SIM

EAP/SIM presents a different view to the customer. Accordingly we start this section with an explanation of the user interface, followed by a discussion of the infrastructure, and an evaluation of EAP/SIM.

### 3.2.1 Use case / Requirements

For EAP/SIM the user needs a SIM card, a card reader and some EAP/SIM client software. Such client software is needed since EAP/SIM is not yet integrated into common operating systems.

The search for an EAP/SIM capable network can be done manually as in the previous use case. Alternatively if supported by the client it is possible to perform an automatic search. After the client has joined the network authentication begins. This happens on layer2 as the client does not yet have an IP address. During authentication the user will be asked for the PIN of his SIM card. After successful authentication the communication between the client and access point is enabled with an encrypted tunnel and the client indicates the successful authentication. Most

EAP/SIM clients show the current status during the authentication. Through this tunnel the client receives an IP address via DHCP. The user can now start the browser and surf through the Internet.

## 3.2.2 Infrastructure

As in the web-based solution we distinguish two cases, a central and a decentral one. For a discussion of the protocol itself see Section 2.5.

### a) Central solution

All packets from the EAP/SIM client on the user notebook get encapsulated by the RADIUS protocol at the access point/authenticator. With the help of this protocol the packets are routed to the RADIUS server. The SSG has an integrated RADIUS proxy to intercept the communication between the access point and the RADIUS server. In some cases this is just a proxy, which sends the packets to the right server. This for example is necessary to support roaming. The RADIUS proxy now selects based on the realm to which RADIUS server the packet to forward. This RADIUS server, using an integrated EAP/SIM plugin, requests the triplets from the HLR or AuC. If the authentication is successful the RADIUS server sends an accept packet to the access point. The access point extracts the session keys for encrypting the wireless interface to the client. And it forwards the EAP accept messages, just as all other EAP and EAP/SIM messages forwarded to the client.

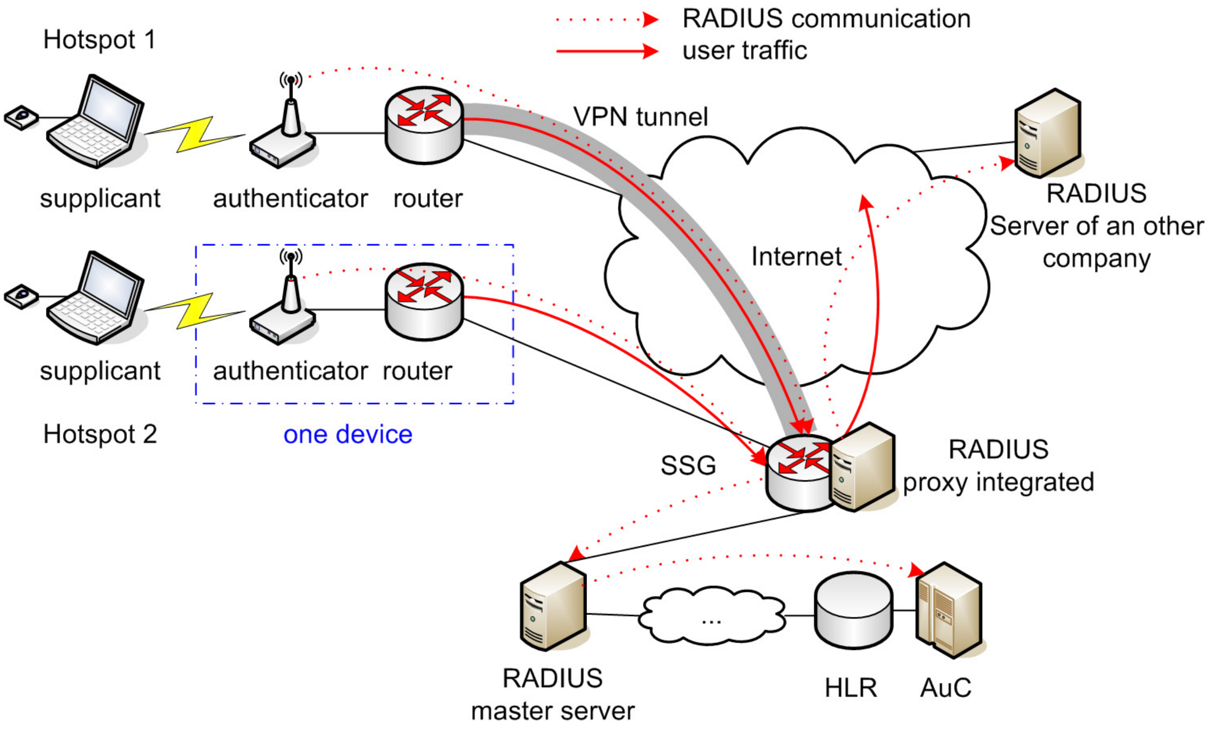


**Figure 3.4.** Typical central EAP/SIM WISP system

The access point now accepts any traffic from the client, but it maybe desired to en- and disable certain services. This cannot be done on the access point but needs to be done on the SSG. But the IP address of the client is not known at the time of authentication, therefore the router sends a RADIUS accounting start packet once the client requests an IP address. This request contains the MAC and IP address of the client. Since the MAC address is already known to the SSG when the authentication was intercepted, the SSG is able to link the IP address to the account used for authentication.

This setup, used by Cisco, causes a strange problem. If a client renews its interface settings, the router sends a RADIUS accounting stop packet followed by a RADIUS accounting start packet. But as the SSG removes all user data upon receiving a stop packet, the start packet is denied and therefore the DHCP request will be rejected.

### b) Decentral solution

All packets from the EAP/SIM client on the user notebook get encapsulated by the RADIUS protocol at the access point/authenticator. In some cases this is just a proxy which sends the packets to the right server. This for example is necessary to support roaming. The RADIUS proxy now selects based on the realm to which RADIUS server the packet to forward. This RADIUS server, using an integrated EAP/SIM plugin, requests the triplets from the HLR or AuC.

If the authentication is successful the RADIUS server sends an accept packet to the access point. The access point extracts the session keys for encrypting the wireless interface to the client. And it forwards the EAP accept messages, just as all other EAP and EAP/SIM messages forwarded to the client. All traffic which arrives through this encrypted tunnel has direct access to the Internet. This is normally done by just routing or bridging the packets to the PAG (Public Access Gateway).

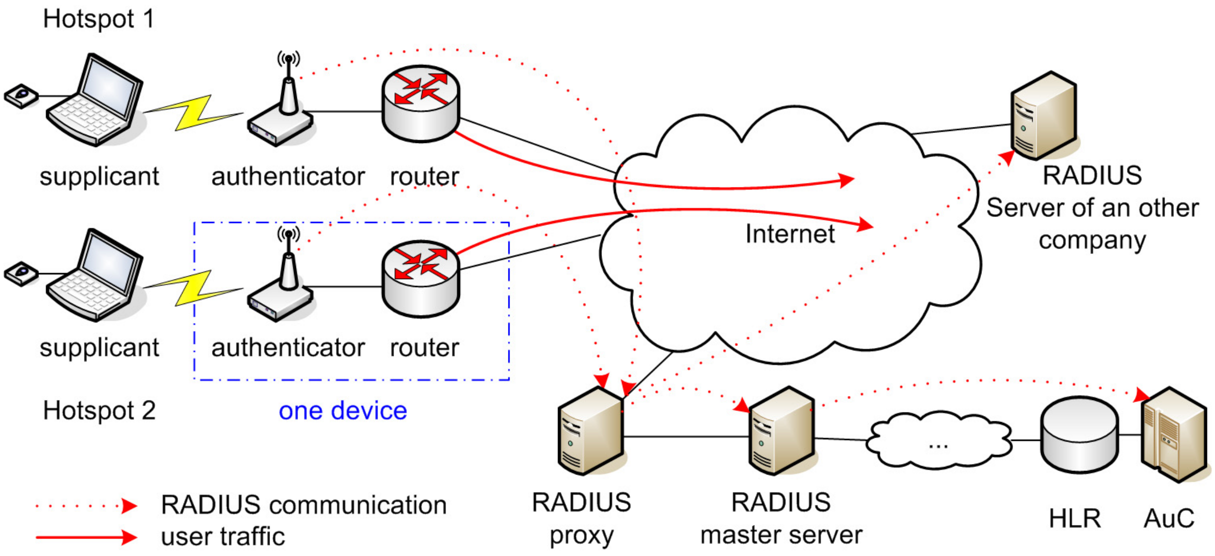


**Figure 3.5.** Typical decentral EAP/SIM WISP system

### 3.2.3 Evaluation

The disadvantages with EAP/SIM do not lie in its security front, this front is in fact an advantage. The disadvantages or problems are in how to migrate the existing hotspots to EAP/SIM capable ones. EAP/SIM provides an air interface encryption and a mutual authentication together with a secret (PIN) easy to remember, but requires special clients and access points. Therefore the access points need to be replaced in existing hotspots in order to migrate to EAP/SIM.

But migrating to EAP/SIM is not always desirable as it may be worthwhile to support both authentication methods. This is a problem as an access point would need to provide an encrypted wireless network for EAP/SIM and one clear text wireless network for web-based authentication. It would then need to pass these separately to the PAG if both are separate devices. But such access points are as rare as 802.1x/EAP capable access points.

If one builds a new hotspot both goals are achievable. Yet these access points will be more expensive than those without such features. For encryption a more powerful CPU is needed, which makes it hard to retrofit old access points even if they're flushable.

An other problem is that intelligence needed at the authenticator must be also managed separately, if it is not the same device as the PAG. One more device to be maintained. For example it needs to know the IP address of the RADIUS server.

# Chapter 4

# Synthesis

The solutions discussed in chapter 3 do have their advantages and disadvantages; EAP/SIM is secure but cannot be easily integrated into existing systems and the web-based methods do not scale in every aspect and are as secure as EAP/SIM.

## 4.1 Conceptual formulation

The ideal solution is a composition of both solutions. The solution should be easily integratable into existing WISP systems and offer a secure and easy way of authentication. It also should support roaming and should be integrateable into the same backend as EAP/SIM and to incorporate with mobile phone operator authentication systems. This implies that the solution needs to use with a RADIUS server, as its the interface the EAP/SIM solution.

In order to support existing deployed WISP systems the solution needs to work without any interaction regarding authentication with the access point - in short it should consider the access point as a transparent device. The solution should be based on GSM SIM cards to increase security if compared with web based solutions. To help the WISP to reduce the costs it should be possible to deploy the server part of the solution (SIM-AS server) in the backend of the WISP. This enables the use of the SIM-AS server at every hotspot or alternately one SIM-AS server for multiple hotspots. This means that the protocol needs to be routeable and therefore must be layer3 or above.

In this combination it is necessary to mention that the task sharing between the PAG and the WISP server differs from system to system - each WISP system vendor follows its own policy. Some Systems are designed for one hotspot, others are for nation-wide WISPs. The solution to be developed should fit into the various existing WISP systems. Changing the PAG may be possible but should be avoided. Afterall it is a big effort to change all PAGs of a WISP. In addition as there are many different WISP systems deployed, each of them would need to be changed to support this solution. This would be a big show stopper.

This implies that the interface to the RADIUS server should be the same as for EAP/SIM so there is no need to change anything in the backend during migration. Both solutions should even run parallel. As companies are a major business target, it needs to be considered that the notebooks are normally administrated by the IT service department. The client software therefore should be able to work without administration right on the notebook.

## 4.2 Principle setup

This section will explain the idea behind the proposed solution which has been implemented in this thesis.

An ideal solution contains the following components:

- Client: The client software needs to communicates with the SIM-AS server and the user.
- PAG: Part of a WISP system and therefore different in every system.
- SIM-AS server: A server component which handles the communication with SIM-AS clients and WISP systems.
- WISP backend: An existing WISP backend from one of the various software companies.

Figure 4.1 shows a possible construction of such a system. The SIM-AS client on the user notebook communicates via a layer3 or above protocol with the SIM-AS server. A new protocol, designed as part of this master thesis, encapsulates EAP/SIM. This means EAP/SIM but not IEEE 802.1x is used, as the SIM-AS client encapsulates EAP/SIM into a routeable protocol before sending the packets directly to the SIM-AS server. The SIM-AS server extracts the EAP/SIM information and forwards it encapsulated in RADIUS to the RADIUS server. Beginning with the RADIUS server, the EAP/SIM authentication chain is untouched. Multi-

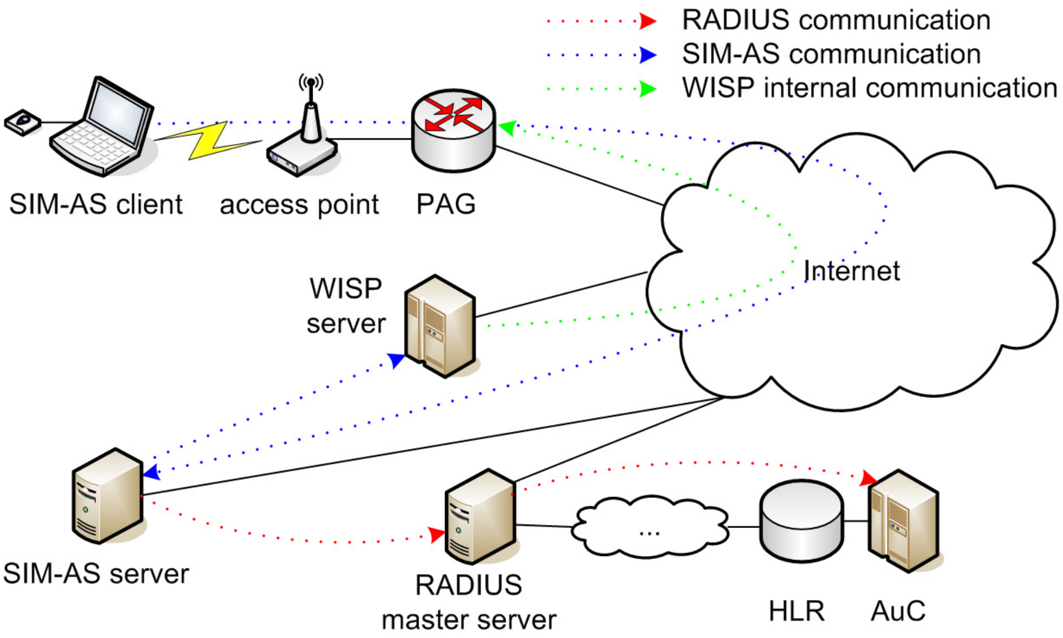


**Figure 4.1.** Typical SIM-AS setup

ple packets are needed for the SIM-AS protocol to carry the EAP/SIM packets, as EAP/SIM requires multiple packets to complete an authentication. If the SIM-AS server receives a RADIUS accept message it contacts the WISP server in order to clear the user at the PAG.

## 4.3 Challenges

As the idea evoled some challenges have appeared:

- How shall the client program communicate with the SIM card?
- How to search for and locate a SIM-AS capable network?
- How to identify the PAG under certain circumstances, where the PAG cannot be identified by its IP address?
- How to address logout and re-authentication encryption problems?
- How to retrieve the data which is necessary to connect to the correct SIM-AS server or how the packets find their way to the server without help from the client, after the client is associated with the wireless network?
- How to communicate with the WISP backend?
- How to handle the communication to the RADIUS server?

These solutions are discussed in separate sections.

## 4.4 Communication with the SIM card

As communicating with the SIM card is needed, a client program for this purpose needs to be developed. To access the SIM card PC/SC is used; this is further discussed in subsection 5.2.1. In this subsection only the client software architecture is discussed.

### 4.4.1 Client software architecture

Choosing a client type has many implications. As some options are only possible with a specific kind of client. The two possible ways for developing a SIM-AS client are: a standalone client or a browser plugin. The company Secartis primary needs a client which runs under Windows, but this does not prevent a platform independent design

#### a) Browser based solution

As the program needs to access the PCSC interface to communicate with the SIM card it cannot run in a sandbox. This means that we either use an ActiveX or a Java Applet that uses JNI and include a native library.

This has following advantages:

- No install medium is necessary, as the installation is done via browser.
- No manual installation from CD-ROM is necessary, which may cause problems for an average user.
- Updates can be made automatically by just putting a new version onto the server.

But there are also the following disadvantages:

- The point with no install medium is only valid if the card reader is supported by the operating system out of the box and no additional drivers are needed.
- The plugin can be browser-dependent, it may even be version dependent. An ActiveX Applet by design runs only with Internet Explorer. A Java Applet would need to access the PCSC interface via a native library which needs to be shipped with the Applet. Also the Java versions shipped with the various browsers are very different.
- During the first use and each update the user would need to accept the download and installation of the program. This window is designed in Windows in such a way that it should and most likely also does scare the average user.
- There is no possibility of searching for a SIM-AS capable network, as the software needs already to be downloaded from a portal site. For this a working connection to the WISP is already necessary.
- Enabling ActiveX is considered a security risk by most IT security experts. Many security related sites describe how to deactivate it [38].

### b) Standalone solution

A standalone client is a normal program, which needs to be installed in the usual way. The disadvantages of these solutions are:

- If required, an automatic update procedure needs to be developed.
- A CD-ROM with the software needs to be shipped to all customer. It may already be outdated at the time of arrival, as CD-ROMs are produced in large quantities. This disadvantages are perhaps not that severe, as the SIM card and the SIM card reader need to be shipped anyway.

Those two disadvantages are juxtaposed to the following advantages:

- The installation onto company notebooks may be done off line by a technician, and the users just need to start the authentication program at the hotspot. As the installation should be done by a user with administrator rights, the normal user of the system does not need to have those.
- No specific browser or version is needed.
- A standalone program is able to search for SIM-AS capable networks.

#### c) Conclusion

The problems with various browsers and versions make it hard to use a browser based solution, specially in a commercial environment, where the IT department takes care of all company computers. Therefore the standalone version has been selected.

## 4.5 SIM-AS capable network detection

It is possible that the client is offered more than one network, on a different or on the same channel. It is an advantage, if the client software associates the client with the appropriate network. The appropriate network in this case is a network, which belongs to his provider or his provider has a roaming agreement with. In this case the operator of the network and therefore the user is able to login with this SIM card. This subsection discusses the possible methods for detecting the appropriate network.

### 4.5.1 SSID

The simplest and fastest method include the marking of the availability of SIM-AS in the SSID of the network, or storing all possible SSIDs in a local database within the SIM-AS client. In the first method the WLAN operator needs to change the SSID of all access points and all roaming partners need to accept the markings of each other or use a standard marking. The second solution requires that a list of all SSIDs of possible hotspots is available and maintained. It is also necessary to update the client database on a regular basis.

### 4.5.2 Domain name

Alternately a domain name supplied to the client by the DHCP server includes a mark to indicate SIM-AS support. This requires the client to send a DHCP request, for which the client needs to be associated with the access point. This makes the system slow and is therefore not useful.

### 4.5.3 Broadcast

The client sends a layer2 broadcast after which the access point, or if the access point is only a bridge, the PAG sends a direct answer to the client. The problem here is that changes on the PAG or access point need to be made. To send layer2 packets the client must be associated with the access point and the SIM-AS client needs to generate the packets on its own. On most operating systems the user needs administrator rights to send broadcast packets.

### 4.5.4 Conclusion

Using SSID is the only suitable solution for quickly finding a SIM-AS capable network. If the access point is not able to send more than one SSID, the primary one needs to be changed. But in most cases the user must find the SIM-AS capable network by himself, which is not such a big drawback as the hotspot network should already have a meaningful name.

## 4.6 PAG identification problem

Under the following circumstances the SIM-AS server needs to identify the PAG, which the client is using, via a PAG ID.

This may be necessary if the IP addresses of the clients are not unique through the whole WISP hotspot system together with dynamic PAG IP addresses. The non-unique IP addresses may occure if different PAGs use the same subnets for the clients combined with masquerading the clients by the PAG. The worldwide reachable IP address of the PAG may change during a session. Therefore the IP address is no longer a unique identifier. Such problems are avoidable in newly designed networks and by some WISP systems, like the Garderos one, even may enforced. Still existing networks have this kind of hotspot partitioning.

One could say that the problem is not that urgent as typically forced a disconnect occurs only after 24 hours online time with (DSL in Germany). Therfore it is unlikely that the disconnection is within the authentication procedure.

There are three flaws in this statement:

- In other countries the forced disconnect occur after a shorter period of time, for example in Austria the typical maximum online time before disconnect is 8 hours.

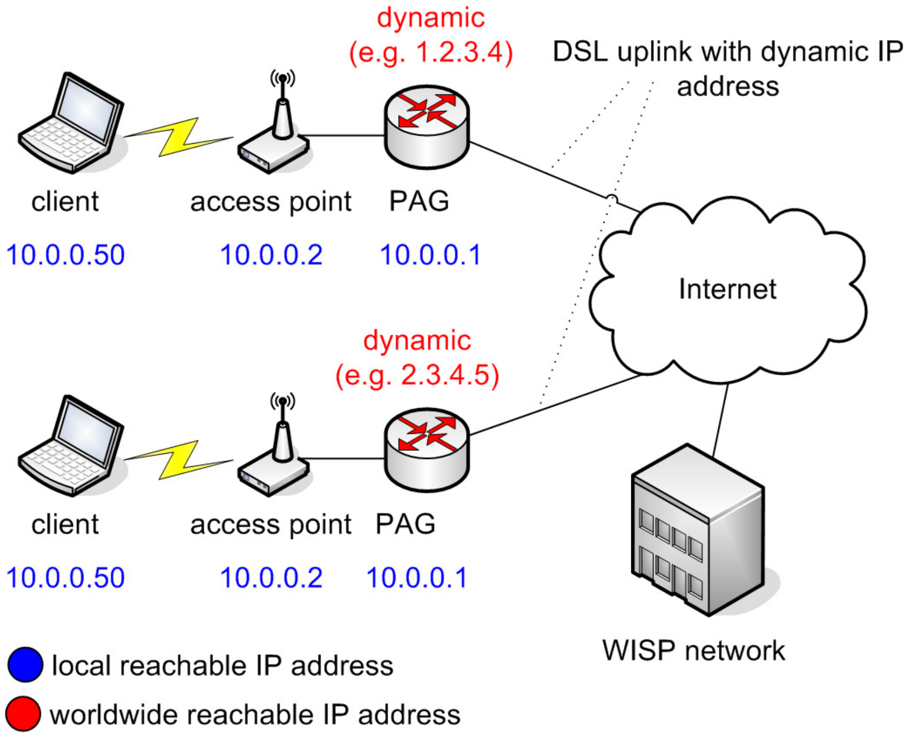


**Figure 4.2.** Identification problem example

- As the user name provided by the client may not be unique, some other data is needed to ensure that the accounting records are unique. Let assume that the user tries to logout but the PAG got a new IP address during the time he was using the hotspot. The SIM-AS server is now unable to remove the user from the logged in database. The user only gets logged out by the idle timeout.
- WISP systems which allow a non unique client IP address handle this internally. But as the SIM-AS server inserts the data into the RADIUS server database, this setup depends on some implementation on the WISP site to set the accounting information for the correct RADIUS user when the user is logged out. For example the WISP needs to store some data to map the current client data in case of a timeout to the old data required to update the clients accounting data.

As the problem cannot be solved in general, the solutions discussed in Section 4.8 need to be aware of this problem.

## 4.7 Logout and re-authentication

This section discusses issues, related to VPN client and intermittent authentication path, which occur in connection with the logout and re-authentication. The VPN client problem applies to all methods of communication with the SIM-AS server. The second problem only applies between the SIM-AS client and the SIM-AS server if connectivity is not always available.

### 4.7.1 VPN client issue

Without a VPN client software the SIM-AS client is able to communicate with the SIM-AS server after successful authentication. Therefore the re-authentication can be forced by the SIM-AS server and an active logout is possible.

If the user uses a VPN client to connect to a companies intranet, most VPN clients, like the Checkpoint IPSEC client, which is used by many companies, block all layer3 traffic not going through the VPN. This ensures that there is no link to the intranet, which bypasses the companies firewall. Even if the user is allowed to use HTTP via the VPN connection, the traffic passes through his employers systems. Therefore it is likely to be filtered, but more important the users visible IP address is from the companies address space. Something that causes trouble for SIM-AS.

As corporate clients are the major client group for a SIM based authentication, this causes the following problems:

- A periodic re-authentication, as with EAP/SIM, is not possible under all circumstances, even if special care is taken to ensure its operation even if the packets travel through the clients company network.
- No layer3 encryption tunnel to the SIM-AS server is realizable under circumstances discussed above.

- An active logout is not possible in every circumstance, even if special care is taken to ensure its operation even if the packets travel through the clients company network. A logout by the WISP is still possible, for example after a predefined time frame without traffic.

### 4.7.2 Intermittent authentication path

An intermittent authentication path is one that is only available as long as the client is not yet authenticated. Two scenarios in which this happens are, if the communication method to the SIM-AS server is only available as long as the client is not authenticated, or if packets are only forwarded by the PAG in this state. Another way how this can arise is if the user closes his browser, in the browser-based authentication solution or if the client crashes.

Several solutions exist to deal with this problem. The first one is to use the current approach of the web-based methods and simply ignore the problem. This is backed up by following statement: "The user has access to the Internet and if he leaves he will be disconnected after an idle timeout anyway.". With a browser based authentication method the ignore part is not such a problem, as the user is not redirected to the login site with the plugin. But if the standalone client is launched and the client shows an error message it cannot communicate with the SIM-AS server, it confuses users.

Therefore some solutions are needed for this problem. They can be separated into client and network site solutions.

#### a) Client site solutions

The first approach is for the SIM-AS client to send a logout packet to the SIM-AS server just before it is closed by the user. But this may not be desirable to the user if he wants to boot another operation system which does not have a SIM-AS client, and it also does not help if the machine crashes.

In another idea the client stores the information, which is needed to connect to the SIM-AS server, on its hard disk. This way the SIM-AS server can still be contacted even if the normal path is not available. For some solutions this is possible without the help from the SIM-AS server. In other cases the SIM-AS server needs to provide the client with the necessary data.

After an abnormal termination of the SIM-AS client the client tries to first connect to the SIM-AS server with the stored data. If it is not possible, the default path will be used. But connecting to the SIM-AS server with the stored data provides a permanent path connecting to the SIM-AS server. Therefore the other way is only needed for retrieving the SIM-AS IP address for the new permanent authentication path method.

#### b) Server site solutions

The only solution which works without support from the SIM-AS client is one where the client can also connect to the SIM-AS server after authentication and therefore one cannot offer an alternative.

#### c) Server and client solution

The SIM-AS client requests a web page it knows is not in the "walled garden" of the WISP. It also knows the content of the requested page, so if the returned web page is equal with the known page content, the client is already authenticated. Otherwise the request is redirected by the PAG to a portal site of the WISP. But this idea only solves the problem of retrieving the status. It does not provide a possibility for logout.

#### d) Conclusion

If the intermittent authentication path problem is one that cannot be ignored, the only clean solution is to use a method for SIM-AS server identification/communication, which ensure permanently availability.

## 4.8 SIM-AS server identification/communication

As the server may not be in the same subnet we need a routeable protocol. In order for the packets to arrive at the SIM-AS server the client needs to know its IP address, otherwise the packets need to be routed by the network to the correct SIM-AS server without the client knowing the SIM-AS IP address. This section shows possible solutions to this problem.

Under certain circumstances the client has to provide the WISP with a unique identification of the PAG, which does not change during the whole accounting time. See Section 4.6.

As there are many aspects and problems to consider with the various SIM-AS server identification solutions, we split the explanation into the following four sections:

- Client
- PAG
- SIM-AS server
- WISP backend

Two DNS methods are conceivable, the first one is an absolute DNS name and the second one a relative method. Both are discussed in separate subsections. The method which works with a broadcast is followed by the HTTP methods which also compose a group of solutions.

### 4.8.1 Absolute DNS name

The client always tries to connect to sim-as.domain.de. To work with more than one SIM-AS server, each DNS server, that the client gets assigned via DHCP, needs to resolve the DNS name to the appropriate SIM-AS server.

## a) Client

### Requirements

- Some DNS server may have to be configured. There are PAGs in some WISP systems that do not force the use of the integrated DHCP server. They intercept just every DNS packet. But if no DNS server is configured the client does not send any DNS request. Therefore it is required that the client uses DHCP or for the WISP systems which does not enforce this the clients needs at least any DNS server configured.

### Implementation work

- A standalone client must be implemented.

## b) PAG

### Requirements

- DNS resolution has to be supported before authentication.
- The PAG needs to permit the access to the SIM AS server IP address for non authenticated users.

### Implementation work

- If its necessary to send a PAG-ID to the client two solutions are thinkable. The first would be a big intrusion into a DNS server by returning different answers depending on the PAG. If each PAG has its own DNS server, not just DNS cache, there are considerably more possibilities to implement this by sending the PAG-ID within a CNAME or TXT field to the client. But a technique needs to be implemented in order to maintain the settings on the PAG.

## c) SIM-AS server

### Requirements

- None

### Implementation work

- None

## d) WISP backend

### Requirements

- None

**Implementation work**

- The DNS servers need to resolve the DNS name of the SIM-AS server.
- A DNS round robin load balancing can be easy implemented.
- If round robin load balancing does not fit the needs, a special DNS server needs to be deployed which supports the needed features. For example special DNS servers exist, which return a different IP address for the same DNS name hinging on the source IP address.
- If every operator uses his own DNS name, every name needs to be entered into each roaming partners DNS.
- Some work is needed to be done to hide and protect the SIM-AS server IP address from the client, as non authenticated users are able to connect to this address.

### 4.8.2 Relative DNS name

As the DHCP answer to the client may also contain a DNS Domain it is possible to use this for relative DNS names such as sim-as.assigned-domain.de. To work with more than one SIM-AS server, the DNS server has to resolve it to the right SIM-AS server.

#### a) Client

**Requirements**

- The client needs to use DHCP, to get the domain in which to resolve the host name.

**Implementation work**

- A standalone client needs to be implemented.

#### b) PAG

**Requirements**

- DNS resolution has to be supported before authentication.
- The PAG needs to permit the access to the SIM AS server IP address for non authenticated users.

**Implementation work**

- None

### c) SIM-AS server

**Requirements**

- None

**Implementation work**

- None

### d) WISP backend

**Requirements**

- None

**Implementation work**

- The DNS servers need to resolve the DNS name of the SIM-AS server.
- If a PAG ID is required, it can be easy provided by using a separate domain name for every hotspot.
- If more than one SIM-AS server must be supported different domains need to resolve to various SIM-AS servers.
- If every operator uses his own DNS name, every name needs to be entered into the roaming partners DNS and as well as into every domain.
- A DNS round robin load balancing may be implemented.
- Additional work is needed to be done to hide and protect the SIM-AS server IP address from the client, as non authenticated users are able to connect to this address.

## 4.8.3 Broadcast

Via broadcast the client can request the IP address of the SIM-AS server. This broadcast must be answered by a system in the subnet. The answer does not have to be a broadcast, as the client address is known. It does not matter whether it is a layer2 oder layer3 packet. In theory it is possible to route a broadcast over subnets, but this requires a VPN to the WISP network as in general the broadcast is not routed over the Internet to the WISP network. Not all WISP systems have a VPN tunnel for their internal communication to the WISP backends.

### a) Client

**Requirements**

- No personal firewall must be installed on the client which blocks the broadcast or the layer2 oder layer3 answer to the broadcast. This is a problem for layer3 packets as they are dropped by default by most personal firewalls, if the connection was not initiated by the client, which is the case here.

**Implementation work**

- A standalone client has to be implemented and the client program must send the broadcast.
- Generating some types of broadcast may be problematicly depending on the operating system and user status.

### b) PAG

**Requirements**

- The PAG needs to be within the range of the broadcast.
- The PAG needs to allow access to the SIM AS server IP address for non authenticated users.

**Implementation work**

- A daemon which listens for these broadcasts must be implemented.
- Some kind of distribution method for the IP address of the various SIM-AS servers needs to be developed. If a DNS name is used to identify the SIM-AS server, the distribution is done via the DNS system.
- If round robin load balancing is required, then this needs to be implemented or if the DNS system is used for distribution, then its load balancing capability can be relied upon.
- If a PAG ID is needed, the daemon can be designed so that it is included in the answer to the broadcast.

### c) SIM-AS server

**Requirements**

- None

**Implementation work**

- None

### d) WISP backend

**Requirements**

- None

**Implementation work**

- Additional work must be carried out to hide and protect the SIM-AS server IP address from the client, as non authenticated users can connect to this address.

## 4.8.4 Port forwarding on the PAG with fix target IP address

The PAG is configured as default gateway for the client and the client connects to a previously defined port on the PAG. The PAG forwards now all packets to the SIM-AS server. The client does not lose a port, which it may use to connect to a server in the Internet, as not every packet at port X will be redirected to the SIM-AS server. It is not essential to use the IP address of the PAG, but it makes life easier.

### a) Client

**Requirements**

- None

**Implementation work**

- A standalone client needs to be implemented.

### b) PAG

**Requirements**

- None

**Implementation work**

- The PAG needs to be changed to forward the packets.
- Some kind of distribution method for the IP address of the various SIM-AS servers needs to be developed. If a DNS name is used for the SIM-AS server the distribution is carried out via the DNS system.

- In order to send something equivalent to a PAG ID to the server, the application layer needs to manipulate the packets from the client.
- If round robin load balancing is required, then this needs to be implemented or if the DNS system is used for distribution, then its load balancing capability can be relied upon.

#### c) SIM-AS server

**Requirements**

- None

**Implementation work**

- None

#### d) WISP backend

**Requirements**

- None

**Implementation work**

- None

### 4.8.5 Port forwarding on the PAG with any target IP address or DNS resolution

This is an obvious approach if the client software is distributed via a browser. The PAG redirects all traffic on port 80 from non authenticated clients to the SIM-AS server, which has a web server with pages containing the client software. The same may be achieved by resolving the DNS name to the SIM-AS IP address upon any request by a non authenticated client. All non port 80 traffic is blocked before authentication is granted.

Normally port 80 redirect is already used by the WISP system. But it is not sufficient as following problems have to be addressed:

- The URL edit field of the browser, e.g. shows "www.in.tum.de", but the content of the browser page is delivered by the SIM-AS server.
- In addition the browser caches the wrong page for this URL.
- If the solution to resolve every DNS request to the SIM-AS server IP address is used, it will pollute the client's DNS cache.

Since these are serious problems this solution is not considered further.

### 4.8.6 Port 80 HTTP redirect on the PAG

This solution attempts to solve some of the problems uncovered above, by sending an HTTP redirect at the PAG instead of just forwarding the packets. A none authenticated client tries to establish a TCP connection to a web server. The PAG redirects the packets to a a daemon running on PAG. After the TCP handshake the client requests a document from the server the client thinks it is connected to. The daemon answers with an HTTP redirect to the portal site of the WISP. The IP address of the WISP portal site is excluded from the redirection rule on the PAG.

This solves the problems with displaying incorrectly the URL in the browser window and browser caching as well as DNS caching. As in the above approach all non port 80 traffic is blocked, before an authentication is granted.

#### a) Client

**Requirements**

- Browser needs to request any site in order to get redirected to the SIM-AS site.

**Implementation work**

- A browser plugin must be implemented.

#### b) PAG

**Requirements**

- The PAG needs to allow access to the SIM AS server IP address also for none authenticated users.

**Implementation work**

- The PAG needs software which carries out the redirect function. A full web server is excessive but a simple daemon which sends the HTTP redirect packet is essential.
- Some kind of distribution method for the IP address of the various SIM-AS server needs to be developed. If a DNS name is used for the SIM-AS server, the distribution is carried out via the DNS system.
- In order to send anything equivalent to PAG ID to the server, the HTTP redirect merely needs to contain a parameter.

### c) SIM-AS server

**Requirements**

- None

**Implementation work**

- A web server needs to be set up on the SIM-AS server.

### d) WISP backend

**Requirements**

- None

**Implementation work**

- Additional work needs to be done to hide and protect the SIM-AS server IP address from the client, as none authenticated user may connect to this address.

### e) Unsolved problem

- As only none authenticated clients are redirected, the user cannot carry out an active logout after a crash.

## 4.8.7 Port forwarding on the PAG at port 80 and HTTP redirect a backend server

The previous solutions works in principle but an HTTP redirect at the PAG is a big change. This solution makes it possible to use the PAG only for packet forwarding.

A none authenticated client tries to establish a TCP connection with a web server. The PAG redirects the packets to a daemon running on a backend server. After the TCP handshake the client requests a document from the server the client thinks it is connected to. The daemon answers with an HTTP redirect to the portal site of the WISP. The IP address of the WISP portal site is excluded from the redirection rule on the PAG. It is likely that the backend server and the portal site are the same machine. Due the support of vhosts in Apache it is also likely that only one HTTP daemon is running.

As with the other solutions all none port 80 traffic is blocked, before authentication is granted. Some parts which are listed under "implementation work" are already available in existing WISP systems, however if not, they need to be implemented and are therefore listed here.

## a) Client

### Requirements

- Browser needs to request any site to get redirected to the SIM-AS site.

### Implementation work

- A browser plugin needs to be implemented.
- If a PAG ID is needed then following solution is possible. As the PAG only forwards the packets, a PAG ID needs to retrieve separately and send then from the client to the SIM-AS server. Retrieving such an ID is possible via the DNS, but in this case requires that the client uses DHCP and that the domain names of the PAGs are unique.

## b) PAG

### Requirements

- The PAG needs to allow access to the SIM AS server IP address also for none authenticated user.

### implementation work

- The PAG only needs to forward port 80 traffic and may change the source IP address. This is an already existing functionality in most operating systems.

## c) SIM-AS server

### Requirements

- None

### Implementation work

- A web server with VHOST support needs to be set up on the machine for serving the sites. The default VHOST simply issues an HTTP redirect to a VHOST with the actual site. The redirect maybe done on another system server of the WISP, so a normal redirect is sufficient.

## d) WISP backend

### Requirements

- None

**Implementation work**

- Additional work needs to be done to hide and protect the SIM-AS server IP address from the client, as none authenticated users can connect to this address.
- If the user should be able to use other authentication methods, a page needs to be presented via which the method can be chosen. This is most likely done by a WISP server and not by the SIM-AS server. This requires no change to the SIM-AS server.
- Load balancing can be realized easily by redirecting to different SIM-AS servers.
- If a PAG ID is required and the solution described for the client is not feasible the web server of the WISP needs to generate a permanent PAG ID from the current IP address of the PAG. This IP address is visible as the PAG forwards and therefore masquerades the client connection. If no masquerading is needed the IP addresses of the clients are unique and therefore no PAG ID generation is needed.

#### e) Unsolved problems

- As only none authenticated clients are redirected the user cannot logout after a crash.

### 4.8.8 Conclusion

The solution with a standalone client and relative DNS name resolution seems to be the easiest to implement and to support. It is also the most comfortable for the customer. Only one problem exists with hotspots where no DHCP is enforced and therefore the relative DNS name solution needs a backup.

A logical choice would be the absolute DNS name solution but that requires that the DNS servers at the hotspots are centrally managed. Central management of DNS servers is required for the PAG ID, if this is not needed a DNS name registered for example at the DENIC is enough and no management or full DNS server at the PAG is needed.

If a PAG ID is needed but only DNS caches are installed on the PAGs then this needs to be changed or another method needs to be chosen as a backup solution. The port forwarding on the PAG followed by an HTTP redirect on a backend server is the logical backup plan for the above described solutions as it is already used at the web-based authentication solutions. If the web server of the portal pages is able to extract a PAG-ID from the forwarding IP address of the PAG, this solution suffices.

For the browser based solution the port forwarding on the PAG followed by an HTTP redirect on a backend server is the most logical choice. The problem with the missing active logout after the browser has been closed is not that important. It provides the features as do web-based authentication methods.

For the prototype the relative DNS name resolution will be used, but an absolute DNS name resolution is also possible without any change on the prototype SIM-AS client.

## 4.9 Communication with RADIUS

The master RADIUS server is the same as for the EAP/SIM solution. Therefore the SIM-AS server needs to act to the RADIUS server as an IEEE 802.1x/EAP capable access point. This adds up to talking to the RADIUS server via RADIUS encapsulated EAP/SIM packets. But there is the problem that under some circumstances the user name which is provided by the access point is not unique, therefore the nasport and nasidentifier and the user name are used by the RADIUS server to generate a unique user ID. The nasidentifier is the IP address of the access point and the nasport is the virtual switch port on the access point. The SIM-AS server does not know the virtual switch port of the client on the access point and therefore needs to generate some alternatives. The following subsections take a look at why and when problems arise and how to circumnavigate them.

### 4.9.1 Not unique username problem

This section discusses under what circumstances the username is not unique and provides solutions to deal with it.

### a) Circumstances

Following three circumstances force a RADIUS server to not use the identity as primary key in the database.

**Invalid identity**

The invalid identity problem is not one that will occur with our client, but it occurs with currently existing EAP/SIM clients. It is possible that, if the solution which is described in this thesis is deployed in a production environment, that other SIM-AS clients show similar behavior. The problem arises since EAP/SIM supports three identities, see 2.5:

- permanent identity (IMSI)
- pseudonym identity (TMSI)
- re-authentication identity

To add further complications, some clients send pseudonyms which forces the RADIUS server to request an other identity. This occurs when using the Windows EAP implementation extended only with EAP/SIM support. The Windows EAP implementation plugin can only store the identity once in the Windows EAP framework. Later Windows always uses this stored identity for answering the identity request packets. If the EAP/SIM plugin now sets an IMSI or TMSI it is not possible to carry out re-authentication as that is not possible with these identities. In order to have the chance to use re-authentication the EAP/SIM plugin needs to provide Windows with an invalid identity. As a consequence the RADIUS server is forced to ask via EAP/SIM for the real identity which is not intercepted by Windows.

An example of such a pseudonym is "invalidId@secartis.owlan.org". As every installation of a client software uses the same identity it is no longer possible to use the identity as a unique user identification at the RADIUS server.

#### Attack on a logged-in user

The EAP identity response packet sends the identity in clear text therefore an attacker is able to retrieve the identity. Once the spied user has been authenticated the attacker sends the EAP identity response packet to the RADIUS. This authentication will fail, but it implies that the real user gets logged out. This would provide the possibility of a denial of service attack.

#### Foreign RADIUS server permanent pseudonym

Different RADIUS servers can assign the same permanent pseudonyms to two clients, for example if they use the same RADIUS software. This creates the same situation discussed above. Especially that the customer does so without realizing it.

### b) Solutions

In this subsection three ideas are presented which try to solve the problems described above without weakening the system.

#### IMSI use only

One may think that using only the IMSI to answer the identity request packet solves the problem. As the Windows EAP plugins cannot be used for our SIM-AS system anyway there is no need to support them. But this weakens the security of the system as it requires that the IMSI is always transmitted. It also does not prevent the denial of service attack on a logged-in user.

#### Database on the SIM-AS server

Another idea is that the SIM-AS holds a table with the identities currently in use and the corresponding IMSI. The EAP identity response itself is not sufficient, but in combination with the EAP-Response/SIM/Start messages it is possible to build this table. But the EAP identity response needs to be sent to the RADIUS server together with a user name before the IMSI can be extracted from the EAP-Response/SIM/Start messages and then be retrieved for the table. We cannot use the EAP identity response for this lookup since it is not unique. The connection between TMSI, the re-authentication identity and the IMSI on the SIM-AS server does not help here.

#### Generate a unique nasport

Since the authentication credentials sent by the client are not sufficient, the nasport and nasidentifier are sent from the access point along with the RADIUS packet. A database on the RADIUS server can use these 3 values as primary key for a user table. The problem with adapting this solution is that we cannot rely on the access point and therefore need to generate some extra data from and/or about the client.

We already need the following values for the WISP system to log the user into the system:

- IP and MAC address of the client
- Visible IP address of the client from the server perspective
- PAG ID if this feature is required.

It is not so easy to generate from these values the needed ones as the type of the variable used for nasport, nasidentifier and username are already specified, as we want to simulate an EAP capable access point to the RADIUS server. The following lines are in the create table call in the used RADIUS server:

- USEDUSERNAME CHAR(128)
- nasport NUMBER
- nasidentifier CHAR(64)

The various possible mapping methods depend on the infrastructure of the WISP, where the SIM-AS system is deployed. Since it is not visible for the SIM-AS client, it can simply be changed from deployment to deployment. In this part only some examples for the various setups are shown.

If the IP address space of the clients is disjunct across all hotspots, the SIM-AS is responsible for a bijective mapping of the client IP address. This works well even if the provider uses upto a class B of IP addresses range.

If the setup is more complicate the nasidentifier also needs to be changed, e.g. to the visible IP address of the client; most likely the external IP address of the PAG. As these changes depend on the WISP infrastructure, in which the SIM-AS system is deployed, we do not discuss it further.

### c) Conclusion

Only the third solution solves all of the discussed problems. The other two are presented to show their limitations. Since generating a unique nasport depends on the environment the SIM-AS system is deployed in, the SIM-AS server should be able to adapt to almost any WISP setup. It needs to be deployable without changes or minimal change.

## 4.10 Communication with the WISP

Communication with the WISP is carried out by the SIM-AS server. Unfortunately the interface differs from WISP system to WISP system. The minimal necessary features of this interface are: user login and user logout. Both are initiated by the SIM-AS server.

The following values need to be transmitted to the WISP server, so that the server can first inform the PAG to forward client packets into the Internet, and second bill the user properly:

- IP and MAC address of the client which need to be provided by the SIM-AS client to the server. Most likely the MAC and IP address are not visible to the SIM-AS server.
- The client IP address which is visible to the SIM-AS.
- Username, nasport, nasidentifier provided at the RADIUS server, which are needed to uniquely identify the user at the RADIUS.
- max_sessiontime and radius_classs which are returned by the RADIUS server after successful authentication

Since the interface differs for WISP system to WISP system two example interfaces developed in this thesis are described in the Chapter 5.

# Chapter 5

# Realization of the SIM-AS server and client

The implementation of the SIM-AS system consists of two parts; the server and client part. The requirements and the corresponding architecture are discussed separately for each part. The chapter concludes with some thoughts around the development and a test environment.

The reader of this chapter, should have the following statment regarding UML diagrams in mind; if a class/interface is used by almost all classes in a package the connection lines are hidden from the figure to keep the diagrams readable.

## 5.1 SIM-AS server architecture

This SIM-AS server architecture section is separated into four parts: requirements, solution architecture, implementation and a short look at the two implemented WISP server interfaces.

### 5.1.1 Requirements

The main goal of the architecture is to be deployable in a WISP system either without or with minimal changes. In addition the following objectives are desirable:

- The SIM-AS server should be maintenance-free. To achieve this the server should be able to run on a system without moving parts. In short the SIM-AS server should be able to run from flash instead of a hard disk.
- The SIM-AS server should be deployable at different locations in the network structure. This ensures the possibility of using a centralized or a decentralized model. For example it should be possible to place the SIM-AS server at the hot spot, in the WISP data center, at the GSM provider, or somewhere in between.

### 5.1.2 Solution architecture

The design of the SIM-AS server reflects these objectives as it supports more than one WISP system at the same time and it is able to decide which WISP system should be contacted and with which parameters. But it is still a simple way to use if such features are not needed.

To reduce maintenance, the server is built statelessly. This is possible as there is no need to keep any session data at the SIM-AS server. The RADIUS server keeps the state. The SIM-AS server only intercepts the accept packet and tells the WISP system to let the client onto the Internet.

Communication to the client is done via SOAP which works over TCP. Communication to the RADIUS server is via UDP. The connections are handled synchronous which means the SOAP function called by the client will return after, for example, the RADIUS server has answered the SIM-AS server request. As it is possible that the RADIUS server does not answer at all, the SIM-AS server resends the UDP requests several times before timing out. The SOAP function will then return and contains an error code for timeout instead of the EAP/SIM packet.

Figure 5.1 shows the chronological flow of packets to and from the SIM-AS server:

1. The SIM-AS client calls a SOAP method of the server. As synchronous SOAP is used, the call returns once the server has finished its work.
2. The EAP/SIM packets which are encapsulated into SOAP are converted to EAP/SIM over RADIUS and send to the RADIUS server. The answer from the RADIUS server is returned as the result value of the SOAP method call.
3. Point 1 and 2 are iterated as long as the RADIUS server does not send a reject or accept packet. The client is informed about the final RADIUS final answer.
4. If the SIM-AS server receives an accept-packet it contacts the WISP server via a WISP system proprietary protocol. Within this protocol the SIM-AS server informs the WISP server of the user's successful authentication.

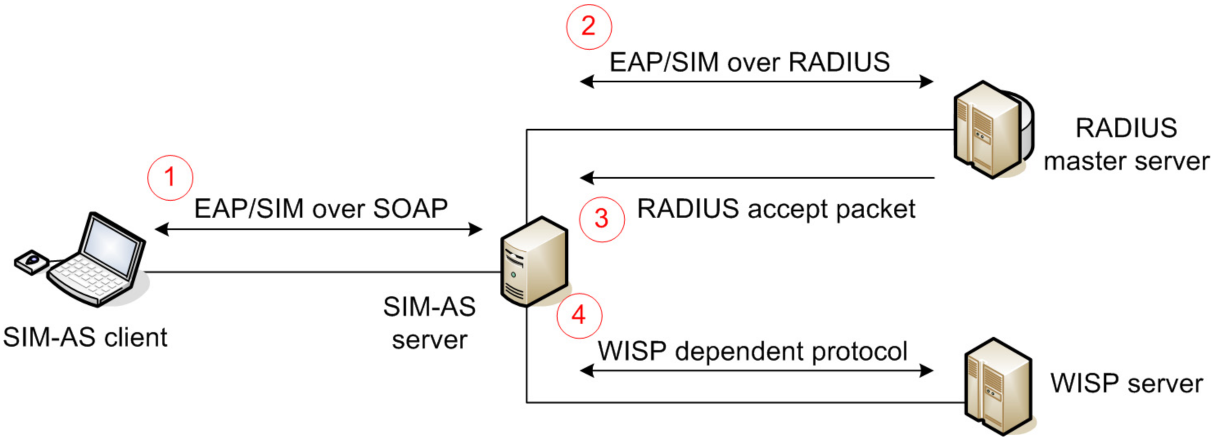


**Figure 5.1.** Chronological flow to and from a SIM-AS server

### 5.1.3 Implementation

This subsection explains how the SIM-AS server is constructed to fulfill this solution architecture. This text does not intend to replace the study of the source code but to give the reader an overview of the architecture.

Figure 5.2 shows a UML diagram that assists in understanding the following explanations of the architecture. As there is no state needed, all objects are created at application server start. This guarantees good performance of the system even under strain. The class which handles the communication to the client is SimAsSoapService. It passes all information plus the visible client IP address from the client via a ClientData object to the SimAsAuthenticationHandler.

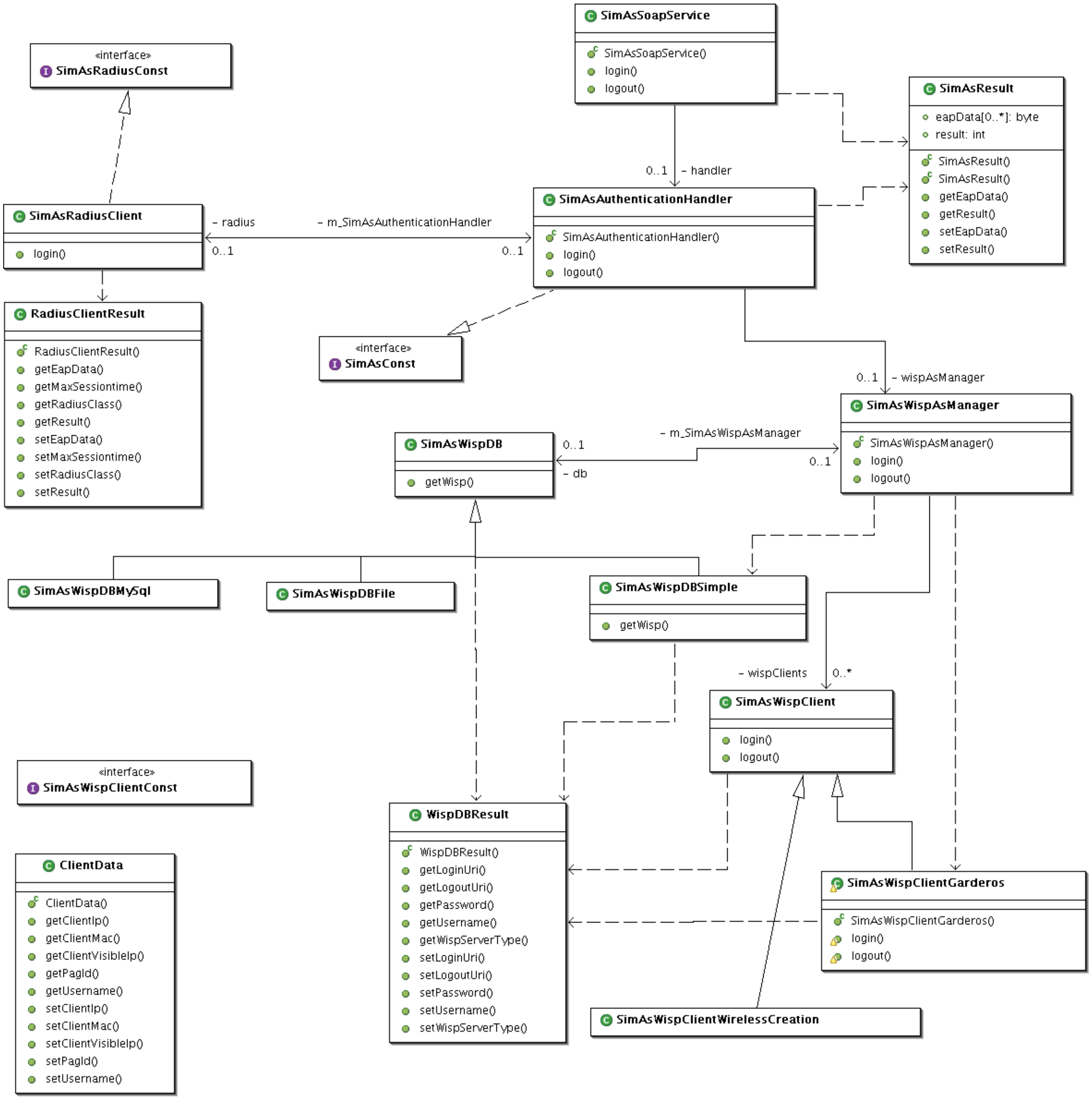


**Figure 5.2.** UML diagram of SIM-AS server

The SimAsAuthenticationHandler sends the EAP frame to the SimAsRadiusClient which handles the communication with the RADIUS. The SimAsRadiusClient returns the answer from the RADIUS to the SimAsAuthenticationHandler which then decides on the next step according to the answer. If it is not an accept, it just returns the EAP/SIM frame to the SOAP handler. If it is an accept, it tells the WISP to let the client into the Internet. This is done using SimAsWispAsManager.

The SimAsWispAsManager asks the configured successor of SimAsWispDB which successor of SimAsWispClient should be called to handle the communication to the WISP. The SimAsWispDB implementation can be a simple one that always returns the same WISP data or it can lookup data in a file or database. This ensures the adaptability of the system to different logical network locations.

Implementations of SimAsWispClient are realized for Garderos and Axiros Wireless Creations. The SimAsWispClient receives an error code from the WISP system or it generates its own error code. This error code is then passed back to the SimAsWispAsManager which sends it via the SimAsSOAPService to the client.

### 5.1.4 Interface to the SIM-AS client

This subsection describes the communication between the SIM-AS client and server. This must be a layer3 (or higher) protocol in order to be routed. The choice fell onto SOAP over HTTPS. It is surely not the only possible protocol, but it is one which simplifies the development.

It is not necessary to create a version handshake for this interface as SOAP already provides the possibility to provide more than one version of a web service at the same time. SimAsSoapService has only two public methods, as it is state-less.

```
public SimAsResult login(SOAPContext soapCtx, String client_ip,
        String client_mac, String username,
        String pag_id, byte[] eap_data )

public SimAsResult logout(SOAPContext soapCtx, String client_ip,
        String client_mac, String username,
        String pag_id, byte[] eap_data )
```

The first parameter SOAPContext is needed to extract the client visible IP address and the IP address that the client is connected to. The object is created by the SOAP classes and is not provided by the client. Table 5.1 shows the various parameters and their values.

### 5.1.5 Interface to the WISP

As part of this thesis, prototype interfaces for Garderos and axiros wireless creations were created. This subsection takes a short look at these interfaces, but does not provide a full interface specification.

### a) Garderos

The first interface which was specified by Secartis and rejected by Garderos was based on SOAP. For Garderos it was too complicated to implement. The interface specified by them is based on HTTPS-get-requests with the values from the SIM-AS server as parameters in the URL. Java build-in http classes were used to implement the interface. In order to support the ’not properly signed’ SSL certificates of the Garderos server some build-in checks needed to be bypassed.

Following parameters are needed for the login method:

- String client_ip
- String client_mac
- String client_visible_ip
- String username
- int nasport
- String nasidentifier
- int max_sessiontime
- String radius_class

Following Parameter are transmitted:

- String username
- int nasport
- String nasidentifier

In both cases the answer from the WISP server is the error code.

| **Parameter** | **Value** |
|---|---|
| client_ip | The client needs to transmit it’s IP address as the PAG may masquerade it and it is therefore not visible to the SIM-AS server. |
| client_mac | Some WISP systems require besides the IP address the MAC address aswell. |
| pag_id | This is a unique identifier of the PAG that the client is connected to. In the prototype this was used for the dynamic DNS name solution. |
| eap_data | Within this array the EAP/SIM packet is carried over the SOAP connection. |
| SimAsResult | SimAsResult is an object with two values. The first is a result code, reporting to the SIM-AS client an error which may have occurred and the second is the array for the EAP/SIM packet from the RADIUS server to the client. |

**Table 5.1.** SIM-AS client and server communication parameters

### b) Axiros wireless creation

The interface to an Axiros WISP server is handled via XMLRPC, which is simular to SOAP. XMLRPC has been chosen as it is used as the internal communication system in the Axiros WISP system. The Apache XmlRpc package has been used to implement the SIM-AS part of the interface.

Following parameters are needed for the login method:

- String client_ip
- String client_mac
- String client_visible_ip
- String username
- int nasport
- String nasidentifier
- int max_sessiontime
- String radius_class

If an error occurres an exception is raised. A logout function has not been defined.

## 5.2 SIM-Client architecture

This section contains four parts. The SIM card reader interface needs to be discussed first, as it has a big impact onto the client's software architecture. After this the requirements of the software architecture are discussed followed by the solution architecture and implementation.

### 5.2.1 Interface to the SIM card reader

In order to communicate with the SIM card, a protocol is needed to communicate with the SIM card reader. The following two interface libraries are commonly used for this purpose.

### a) CT-API

The CT-API standard is widespread in Germany, it provides an easy interface to the SIM card but it is not shipped with Windows.

### b) PC/SC

The PC/SC Specifications 1.0 were released in 1997, and the PC/SC Workgroup is working on version 2.0. PC/SC support is integrated into Windows 2000/XP and it is used by every EAP/SIM solution on the market. Also all major SIM card reader producers support PC/SC.

### c) Conclusion

The choice of PC/SC was not difficult. PC/SC is supported by Windows and is also used by all known EAP/SIM solutions.

## 5.2.2 Requirements

The client is more complicated than the server, as the client cannot be built stateless. It needs to keep track of the authentication state. The following implementation requirements need to be fulfilled:

- Parse and generate the EAP/SIM packets
- Communicate with the SIM card.
- Handle the communication with the user.
- Reuse individual parts of the client in other projects.

## 5.2.3 Solution architecture

To meet the requirements mentioned above the client was separated in two pieces, the GUI and the backend part. Both communicate via a well defined interface, which is designed in a way that the GUI can be changed without any change to the backend. It even supports a console user interface as will be seen later. Figure 5.3 shows the main components of the client architecture.

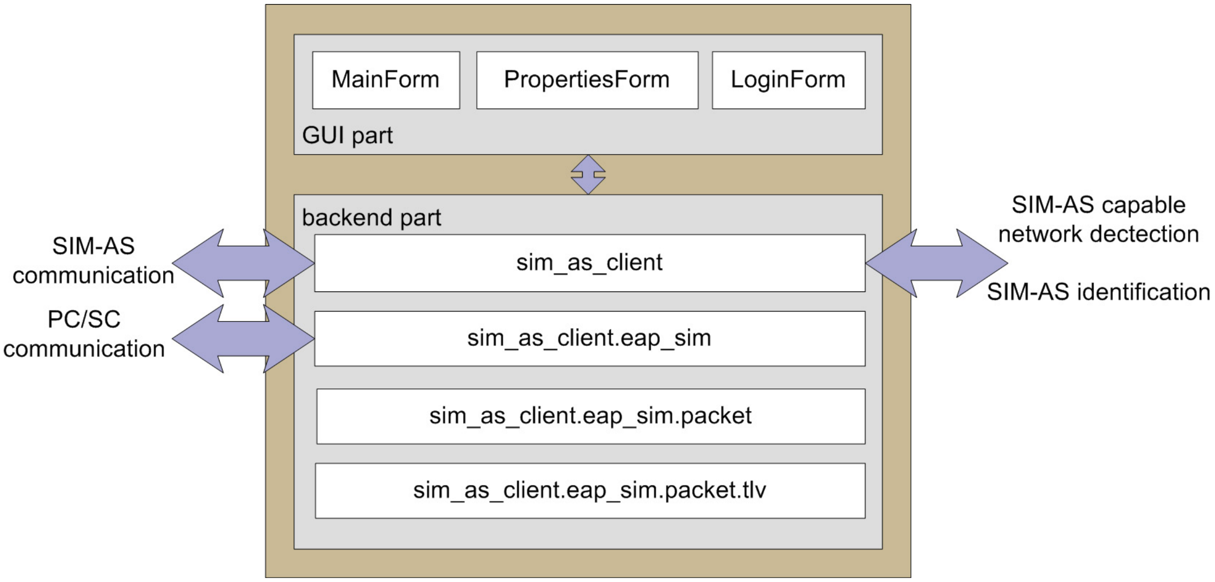


**Figure 5.3.** SIM-AS server overall architecture

### a) GUI

The graphical user interface consists of three windows. The main window contains, see Figure 5.3, 4 buttons and a status field. The properties button leads to the properties window shown in Figure 5.4. The PIN window is used if the user needs to enter the PIN for his SIM card.

### b) Backend

Concerning class design the client backend is the most complexe structure, as it needs to implement the EAP/SIM protocol and therefore needs to communicate with the SIM card. To deal with this complexity the backend is separated into the following packages:

#### sim_as_client package

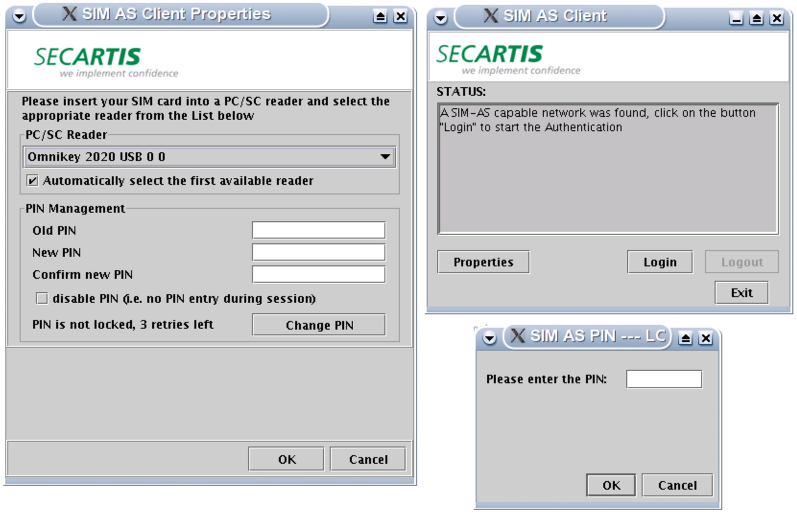
This package handles the communication with the GUI and with the SIM-AS server. It is also responsible for finding a SIM-AS capable network and extracting the information necessary to connect to the SIM-AS server.

#### sim_as_client.eap_sim package

This package contains the EAP/SIM classes for statehandling. This implies that they are persistent during the whole authentication and that they store the current authentication state.

#### sim_as_client.eap_sim.packet package

In contrast to the previous package this contains the classes to handle individual requests. These use the classes of the sim_as_client.eap_sim package to manipulat the state. This package is responsible for parsing and generating EAP/SIM packets.



Please insert your SIM card into a PC/SC reader and select the appropriate reader from the List below

PC/SC Reader: Omnikey 2020 USB 0 0

☑ Automatically select the first available reader

Old PIN

New PIN

Confirm new PIN

□ disable PIN (i.e. no PIN entry during session)

PIN is not locked, 3 retries left

A SIM-AS capable network was found, click on the button "Login" to start the Authentication

Please enter the PIN:

**Figure 5.4.** Screenshots of SIM-AS client

### sim_as_client.eap_sim.packet.tlv package

TLV stands for Tag Length Value and all classes in this package represent a tag in the EAP/SIM protocol. They are used to parse, store and generate these tags.

## 5.2.4 Implementation

This section dives deeper into the detail of implementation of the SIM-AS client. Note that only a prototype has been implemented. Therefore not all discussed solutions in chapter 4 are implemented. For the SIM-AS server identification the relative and absolute DNS name methods have been chosen. In the dynamic DNS name version the domain name was used as PAG ID. Therefore the temporal authentication path problem is not an issue. The SIM-AS server capable network detection has not been implemented.

### a) GUI

As shown in the class diagrams (Figure 5.5) a window always consists of a WindowName and WindowNameExtd class. The purpose of this arrangement is to separate the GUI design from the code, which contains the intelligence and communicates with the backend. The WindowName class is created by the visual GUI designer and the WindowNameExtd is the successor to it. The GuiHelper class enables the three windows to center on the screen.

### b) Backend

The backend section is split across the following packages.

### sim_as_client package

The SimAsClient class handles communication to the GUI. This package also consists of the classes that implement methods to find a SIM-AS capable network and retrieve the data necessary to connect to the SIM-AS. SimAsSoapClient is used by SimAsClient to communicate with the SIM-AS server. SimAsClient passes the EAP/SIM frame to its sub-package

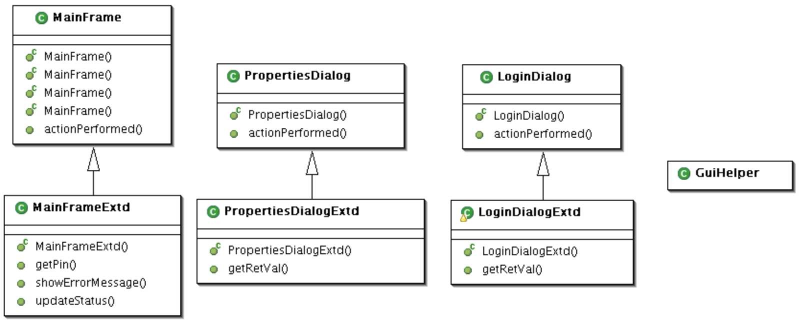


**Figure 5.5.** UML diagram of the GUI part of the SIM-AS client

sim_as_client.eap_sim.

As the GUI needs to display the status updates from the backend, the backend needs to run non-blocking realised via thread. But it should also be possible to use the backend in blocking mode. The SimAsClient is the interface class for the GUI and it implements Runnable and SimAsClientConst. The GUI calls the SimAsClient as thread with the following two lines

```
Thread login = new Thread(new SimAsClient(this, THREAD_LOGIN));
login.start();
```

The thread is now started. To keep the surface up to date on the status of the authentication MainFrameExtd implements SimAsCallBack and SimAsClientConst. As shown in the above code sample, the SimAsClient constructor takes the calling GUI object as parameter and expects the following methods, defined in Interface SimAsCallBack:

- void updateStatus(int status) is called to inform the user about of the current state.
- String getPin() is called to retrieve the PIN. If cancel is pressed by the user, it returns null as String
- void showErrorMessage(String s) is called following a critical error that forces the SimAsClient class to terminate. It displayS an error message window to the user.

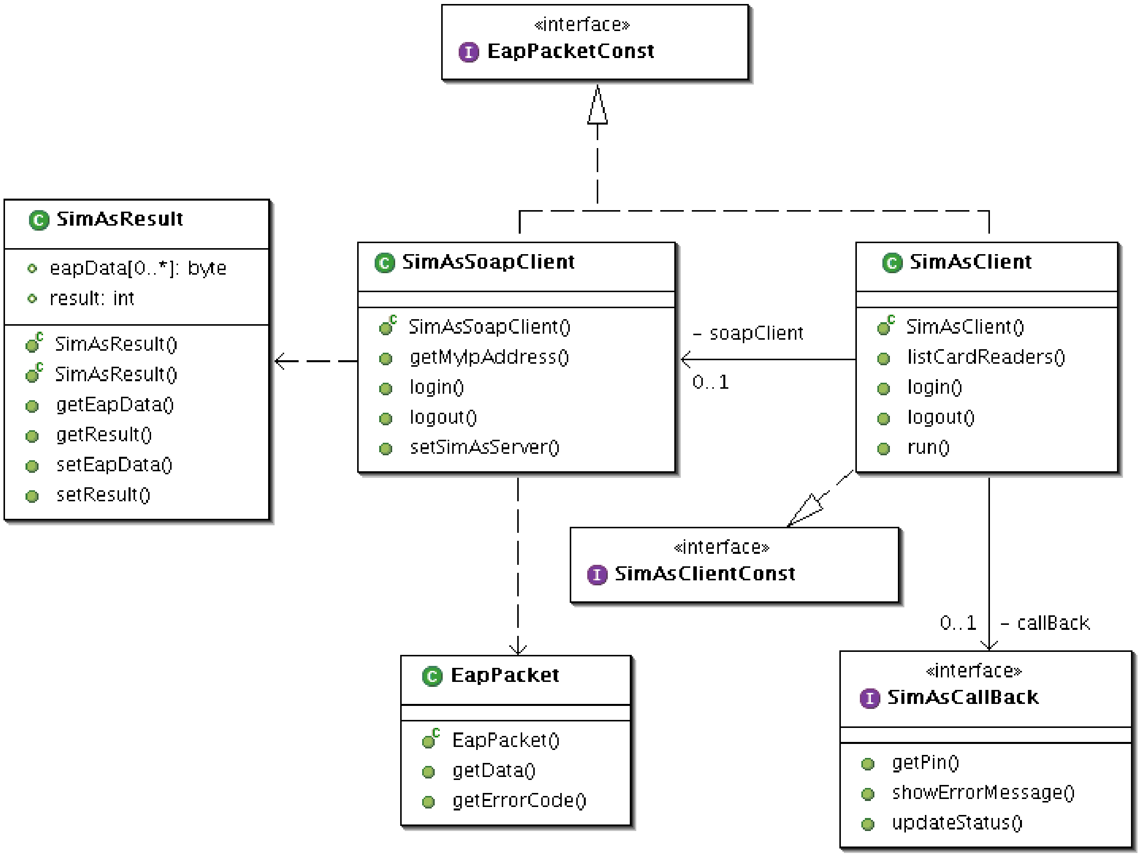


**Figure 5.6.** UML diagram of the sim_as_client package

*Communication to the parent package*

The communication for this package from the sim_as_client is handled by EapSimHandler with the two main methods getIdentityPacket(), which returns the frame needed to start the communication, and handlePacket(), to which all returned frames from the SIM-AS are given. The method handlePacket() processes a packet and determins the answer packet.

### sim_as_client.eap_sim

EapSimHandler is the interface class for the parent package sim_as_client and provides two main methods. The first returns the EAP identity response packet and the second handles an incoming packet and returns the corresponding response packet and some extra information for the calling class. Within the handle method the EapSimPacket class of the subpackage sim_as_client.eap_sim.packet is used to parse the packet.

EapSimState provides an interface to all persistent values which are needed during the authen-

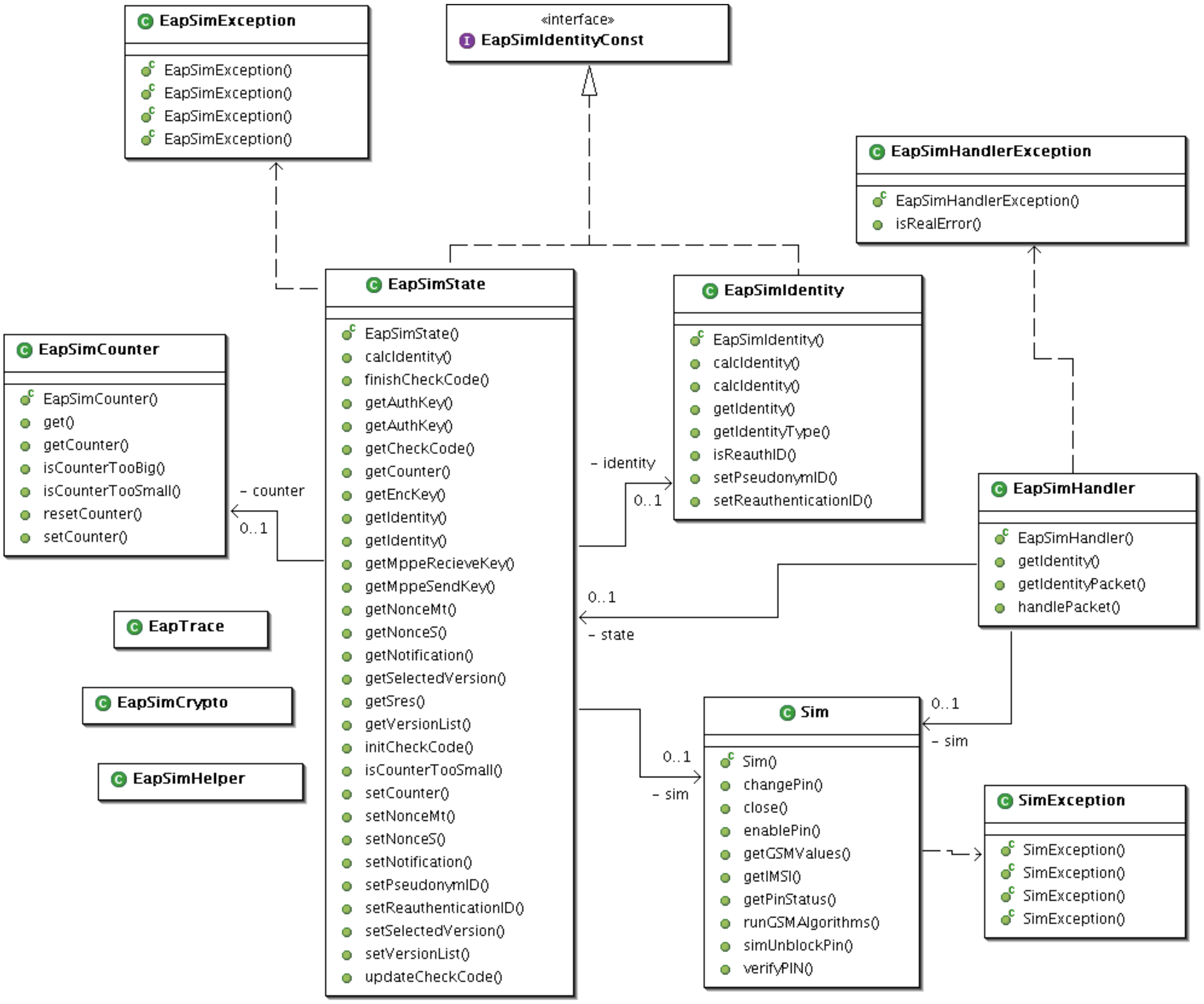


**Figure 5.7.** UML diagram of the sim_as_client.eap_sim package

tication. Some values are just stored in the EapSimState object. Others need to be calculated at request of EapSimState, which may distribute the workload on classes such as EapSimIdentity, EapSimCounter and Sim. EapSimIdentity calculates the correct identity in response to the server identity requests. EapSimCounter checks and generates the correct packet IDs used in the EAP protocol. Sim encapsulates all SIM card related work, by providing easy methods such as getIMSI().

### sim_as_client.eap_sim.packet

The EapSimPacket class provides two constructors, one for parsing EAP/SIM packets and the other for generating EAP/SIM packets. The packets to be parsed are provided as a byte array to the constructor. The constructor splits the package into tags which are then used as parameters to the constructors of the TLV classes. After that the provided tags and the type of the packet is compared within EapSimDefindedPackets to check if it is a valid EAP/SIM packet. The method verifyMACandDecrypt() is used to check the MAC of a packet.

The EAP/SIM generating constructor is called with the EapSimPacket object which parses the EAP/SIM packet. It uses EapSimDefindedPackets to retrieve a template of the EAP/SIM packet which is then filled. Finally makeSendStruct() is used to generate a byte array of the full EAP/SIM packet.

EncrDecrTLSData is used to en- and decrypt parts of a packet. EapSimHeader is used to generate and verify the EAP/SIM header. EapSimPacketType does the same for the type of a packet.

### sim_as_client.eap_sim.packet.tlv

This package contains all tag classes necessary for the EAP/SIM protocol. All are successors to the TLVBaseElement class. For some tags two bytes have been reserved for later use, in the class hierarchy these are successors to TLVBaseWithReserved. The various identification tags are successors of TLVBaseId.

TLVBaseElement implements almost all methods needed to access the tags getLength(), getTag(), getValue() and toByteArray().

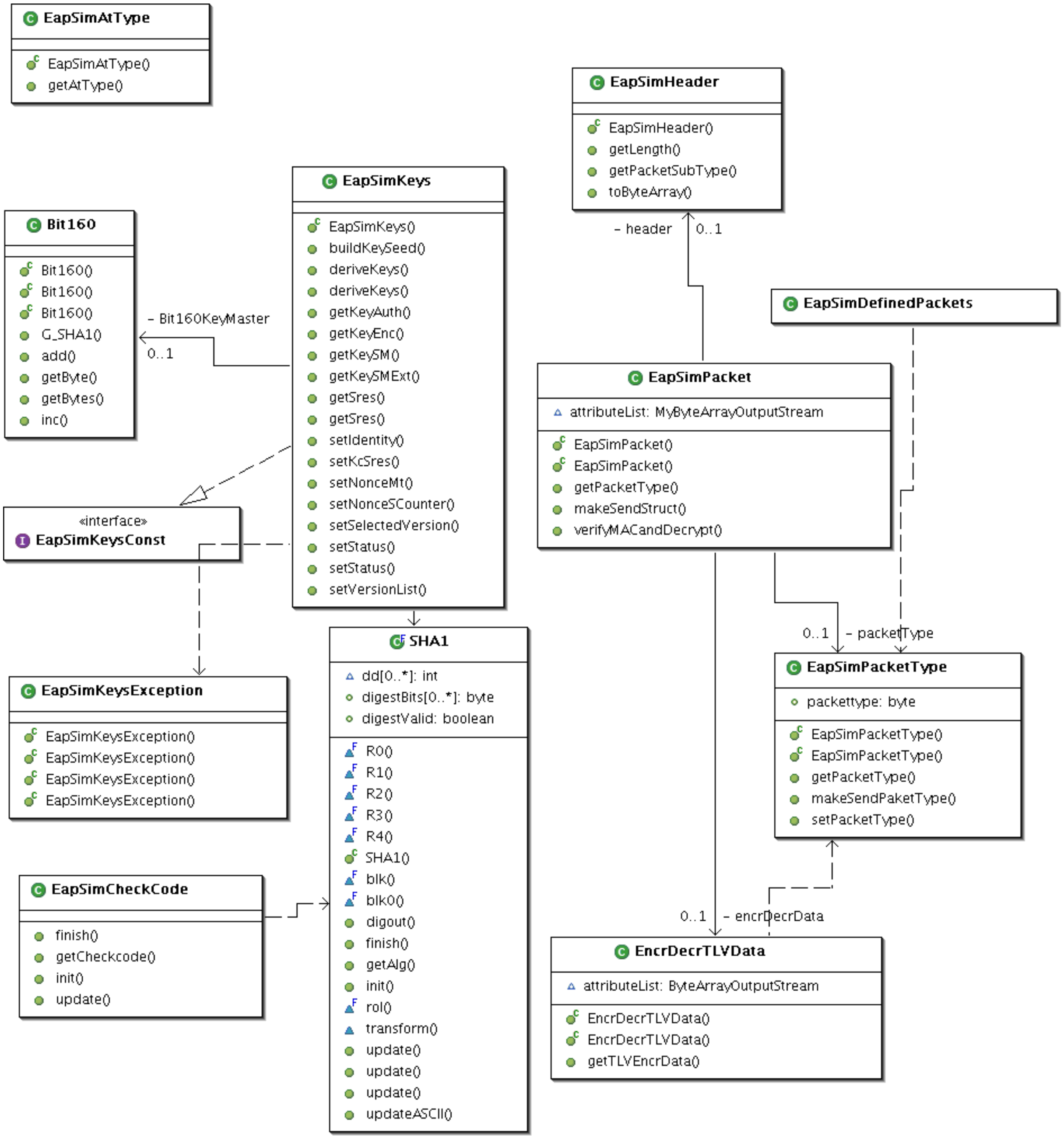


**Figure 5.8.** UML diagram of the sim_as_client.eap_sim.packet package

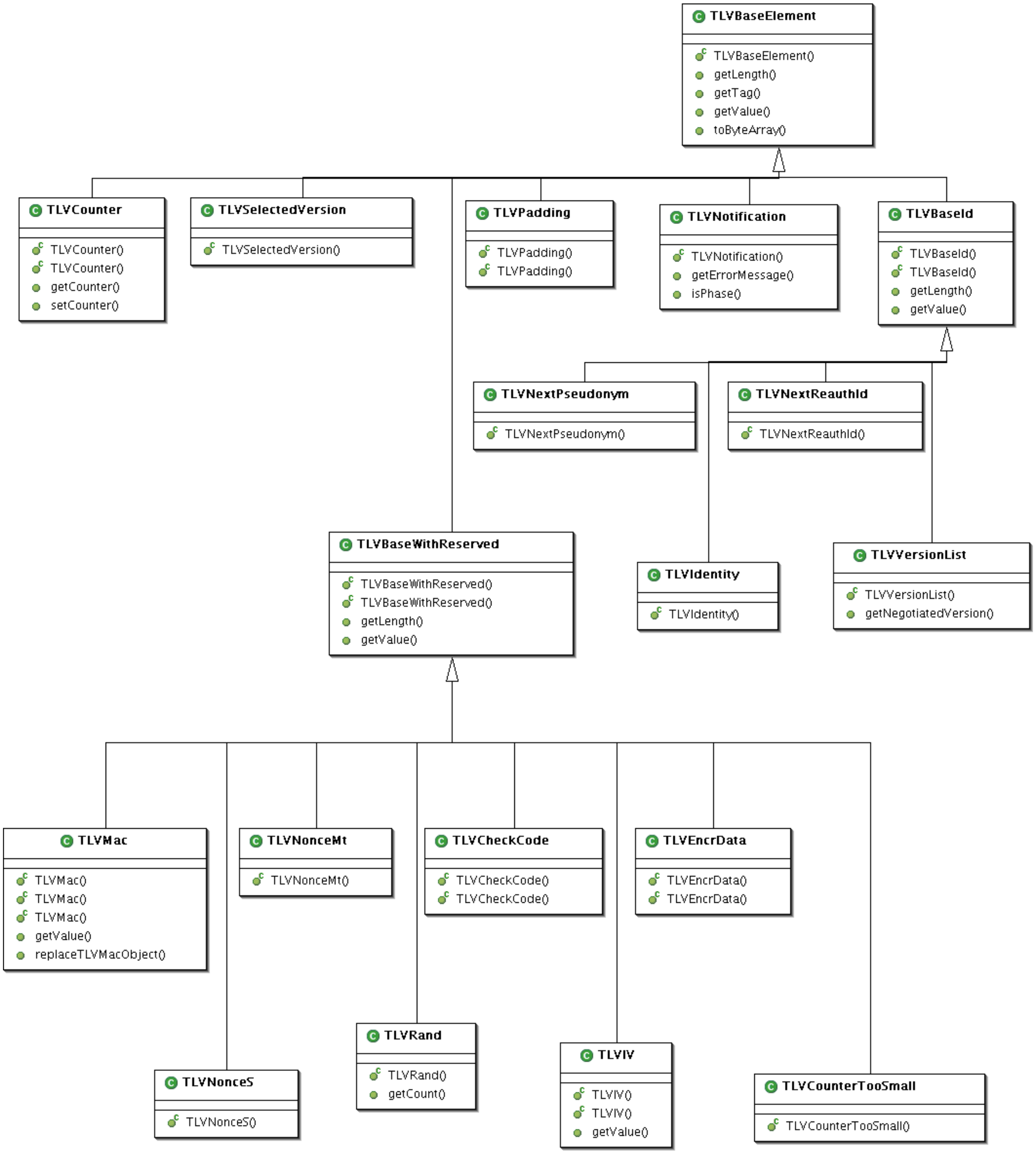


**Figure 5.9.** UML diagram of the sim_as_client.eap_sim.packet.tlv package

## 5.3 Development environment

Initially the SIM-AS server and the SIM-AS client were developed and tested on the same computer. This had to be changed as soon as integration tests with WISP system vendors were conducted. Now the SIM-AS server had to communicate with test servers at their data centers. From this point onward the SIM-AS server was deployed on a Debian-Woody Linux server. The server was connected directly to the Internet with no firewall system between it and the Internet. Later in the development process for ease of debugging a RAIDUS server and an HLR simulator were installed on the same machine. The Perl implementation Radiator 3.9 with the RADIUS-EAP-SIM 1.8 extensions was used as RADIUS server, as it is the standard server depolyed by Secartis. Radiator is also popular with telecoms and WISPs in production environments.

The SIM-AS client development machine was a computer with SUSE 8.2, but it was also tested on a Windows XP notebook in regular intervals, to ensured the platform independently of the client. In the beginning Java 1.4.1 was used as shipped with SUSE. Later it needed to be changed to 1.4.2 in order to support AES.

### 5.3.1 Integrated Development Environment (IDE)

As IDE, Eclipse 2.1 was chosen since it is freely available and powerful. The availability of many plugins and the good outdoor's knowledge of Eclipse helped in the decision process.

The following plugins were used in this thesis:

- Graphical Editing Framework from eclipse.org to create the graphical surface. (http://www.eclipse.org/gef/)
- Eclipse UML from Omondo was used to create UML class diagrams. (UML –> Code, Code –> UML) (http://www.omondo.com/)

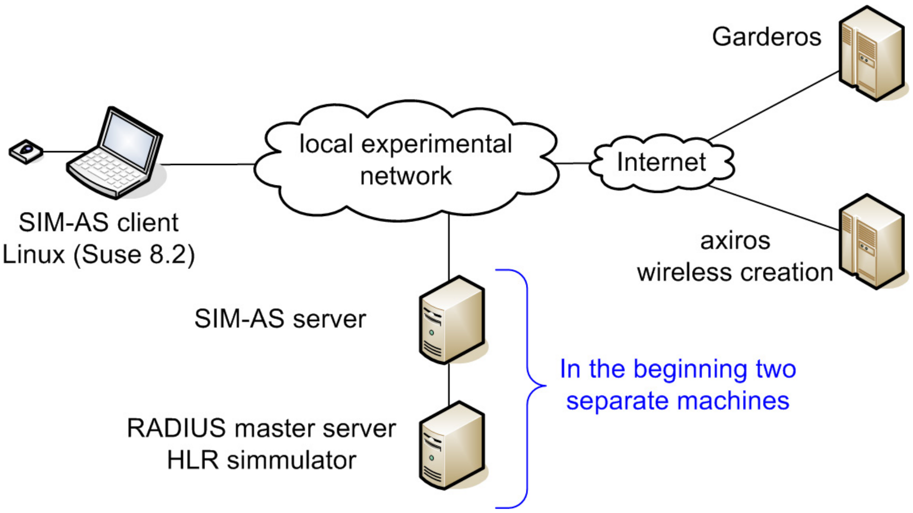


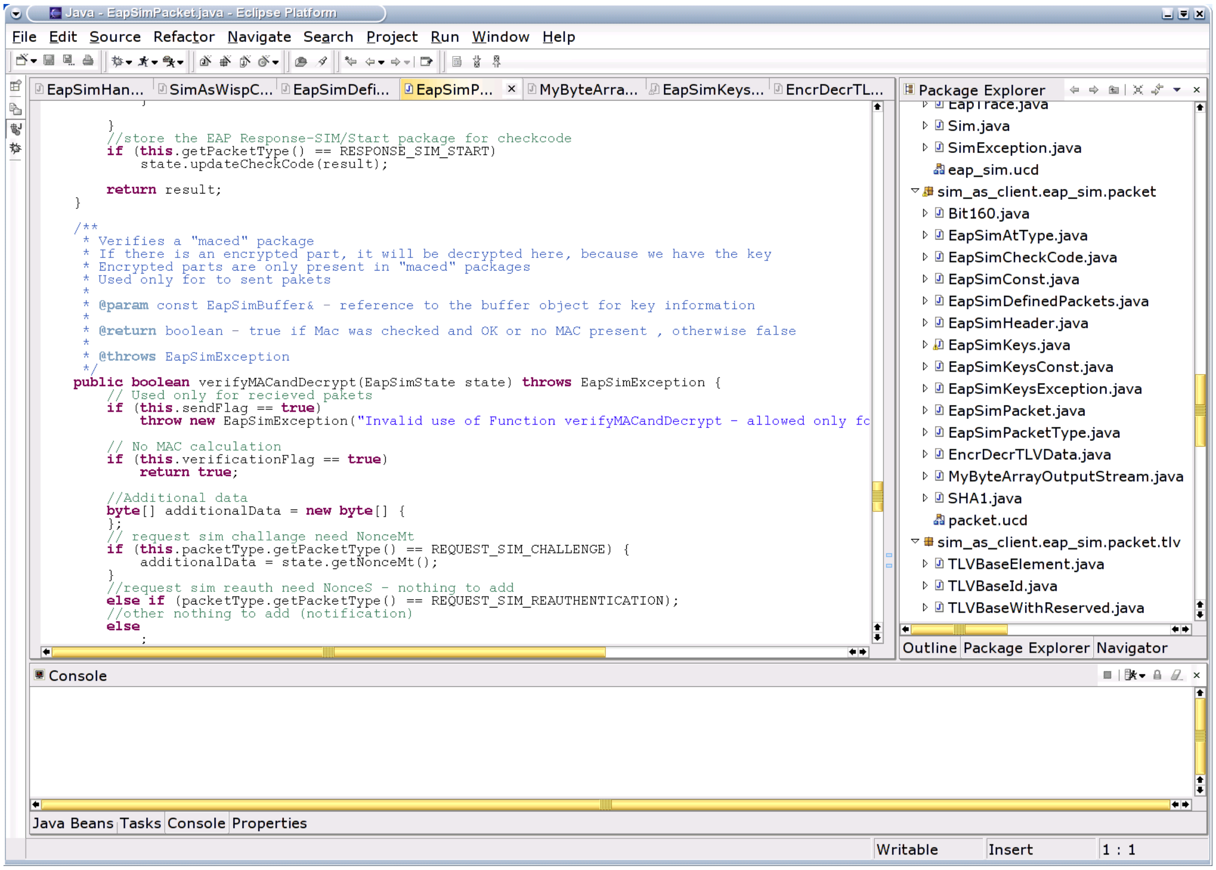
**Figure 5.10.** Development environment setup

## 5.3.2 Libraries

This section describes which libraries where used and for which purpose. For SIM-AS server the RADIUS Client Library (http://www.axlradius.com/RadiusClient.htm) from Michael Lecuyer was used. Jakarta Tomcat 5.0.18 was used as development application server. To support SOAP and XMLRPC the SOAP and XMLRPC libraries from Apache.org are used.

For the SIM-AS client additional libraries are used. Starting from the SOAP library from Apache.org for connecting to the server to PC/SC for communicating with the SIM card. IBM provides a Java implementation to encapsulate this interface. This encapsulation is currently available for Linux and Windows. In addition a Base64 and a SHA1 class are used, which were provided by Secartis.

At the beginning it was intended to use the IAIK Java Cryptography Extension from TU-Graz. This was not necessary as Java 1.4.2 needs an "Unlimited Strength Jurisdiction Policy" patch in order to use the IAIK extension. But this patch already contains the AES algorithms and as performance is not an issue, the AES from this patch was used.



```
        }
        //store the EAP Response-SIM/Start package for checkcode
        if (this.getPacketType() == RESPONSE_SIM_START)
            state.updateCheckCode(result);

        return result;
    }

    /**
     * Verifies a "maced" package
     * If there is an encrypted part, it will be decrypted here, because we have the key
     * Encrypted parts are only present in "maced" packages
     * Used only for to sent pakets
     *
     * @param const EapSimBuffer& - reference to the buffer object for key information
     *
     * @return boolean - true if Mac was checked and OK or no MAC present , otherwise false
     *
     * @throws EapSimException
     */
    public boolean verifyMACandDecrypt(EapSimState state) throws EapSimException {
        // Used only for recieved pakets
        if (this.sendFlag == true)
            throw new EapSimException("Invalid use of Function verifyMACandDecrypt - allowed only fc

        // No MAC calculation
        if (this.verificationFlag == true)
            return true;

        //Additional data
        byte[] additionalData = new byte[] {
        };
        // request sim challange need NonceMt
        if (this.packetType.getPacketType() == REQUEST_SIM_CHALLENGE) {
            additionalData = state.getNonceMt();
        }
        //request sim reauth need NonceS - nothing to add
        else if (packetType.getPacketType() == REQUEST_SIM_REAUTHENTICATION);
        //other nothing to add (notification)
        else
            ;
```

**Figure 5.11.** Eclipse snapshot

### 5.3.3 Debugging Environment

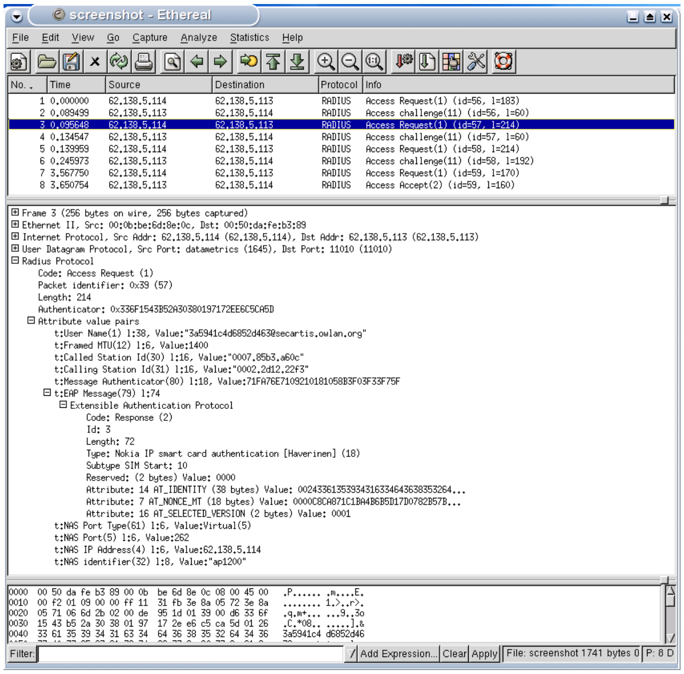
The following environment was used to debug the client and server. On the client almost all debugging work was done with debug messages to the console. An extra class was implemented for this work. Exceptions were dealt with printing a trace to the console. To check the received and sent packets in an easier way a patched ethereal version was used that is able to show the tags of an EAP/SIM packet as shown in Figure 5.12.

This ethereal version is able to work with Layer2 EAP/SIM packets, which was not needed in our case, and RADIUS packets containing EAP/SIM frames. The later was used to debug the messages between the SIM-AS and the RADIUS server. This is sufficient since the SIM-AS server passes the EAP/SIM packets one to one from the client to the RADIUS.

During this thesis it was possible to access and modify the source of the RADIUS server to insert debug output in order to compare it with data of the client and the ethereal output.

### 5.3.4 Test environment

The purpose of our test environment is to resemble the target environment as much as possible. To achieve this the SIM-AS server was moved on a VIA EPIA board system which is runable from flash. The client has been deployed on a Windows XP notebook with an integrated WLAN card. As access point we used a Cisco 1200, which is also able to handle EAP/SIM. An access point without 802.1x support would have been sufficient. The access point was configured in bridging mode and connected to the PAGs from Garderos and Axiros. These devices provided a



```
No. .  Time      Source         Destination    Protocol  Info
     1 0.000000  62.138.5.114   62.138.5.113   RADIUS    Access Request(1) (id=56, l=183)
     2 0.089499  62.138.5.113   62.138.5.114   RADIUS    Access challenge(11) (id=56, l=60)
     3 0.095648  62.138.5.114   62.138.5.113   RADIUS    Access Request(1) (id=57, l=214)
     4 0.134547  62.138.5.113   62.138.5.114   RADIUS    Access challenge(11) (id=57, l=60)
     5 0.139959  62.138.5.114   62.138.5.113   RADIUS    Access Request(1) (id=58, l=214)
     6 0.245973  62.138.5.113   62.138.5.114   RADIUS    Access challenge(11) (id=58, l=192)
     7 3.567750  62.138.5.114   62.138.5.113   RADIUS    Access Request(1) (id=59, l=170)
     8 3.650754  62.138.5.113   62.138.5.114   RADIUS    Access Accept(2) (id=59, l=160)
```

```
⊞ Frame 3 (256 bytes on wire, 256 bytes captured)
⊞ Ethernet II, Src: 00:0b:be:6d:8e:0c, Dst: 00:50:da:fe:b3:89
⊞ Internet Protocol, Src Addr: 62.138.5.114 (62.138.5.114), Dst Addr: 62.138.5.113 (62.138.5.113)
⊞ User Datagram Protocol, Src Port: datametrics (1645), Dst Port: 11010 (11010)
⊟ Radius Protocol
    Code: Access Request (1)
    Packet identifier: 0x39 (57)
    Length: 214
    Authenticator: 0x336F1543B52A30380197172EE6C5CA5D
  ⊟ Attribute value pairs
      t:User Name(1) l:38, Value:"3a5941c4d6852d463@secartis.owlan.org"
      t:Framed MTU(12) l:6, Value:1400
      t:Called Station Id(30) l:16, Value:"0007.85b3.a60c"
      t:Calling Station Id(31) l:16, Value:"0002.2d12.22f3"
      t:Message Authenticator(80) l:18, Value:71FA76E7109210181058B3F03F33F75F
    ⊟ t:EAP Message(79) l:74
      ⊟ Extensible Authentication Protocol
          Code: Response (2)
          Id: 3
          Length: 72
          Type: Nokia IP smart card authentication [Haverinen] (18)
          Subtype SIM Start: 10
          Reserved: (2 bytes) Value: 0000
          Attribute: 14 AT_IDENTITY (38 bytes) Value: 00243361353934316334643638353264...
          Attribute: 7 AT_NONCE_MT (18 bytes) Value: 0000C8CA871C1BA4B6B5D17D0782B57B...
          Attribute: 16 AT_SELECTED_VERSION (2 bytes) Value: 0001
      t:NAS Port Type(61) l:6, Value:Virtual(5)
      t:NAS Port(5) l:6, Value:262
      t:NAS IP Address(4) l:6, Value:62.138.5.114
      t:NAS identifier(32) l:8, Value:"ap1200"
```

```
0000  00 50 da fe b3 89 00 0b  be 6d 8e 0c 08 00 45 00   .P...... .m....E.
0010  00 f2 01 09 00 00 ff 11  31 fb 3e 8a 05 72 3e 8a   ........ 1.>..r>.
0020  05 71 06 6d 2b 02 00 de  95 1d 01 39 00 d6 33 6f   .q.m+... ...9..3o
0030  15 43 b5 2a 30 38 01 97  17 2e e6 c5 ca 5d 01 26   .C.*08.. .....].&
0040  33 61 35 39 34 31 63 34  64 36 38 35 32 64 34 36   3a5941c4 d6852d46
```

**Figure 5.12.** Ethereal snapshot of a sniffed RADIUS packet encapsulating an EAP/SIM packet

DHCP server and blocked the communication to the Internet. But they allowed communication between the SIM-AS client and SIM-AS server for authentication.

This setup is illustrated in Figure 5.13 and it resembles a real live hotspot relatively well.

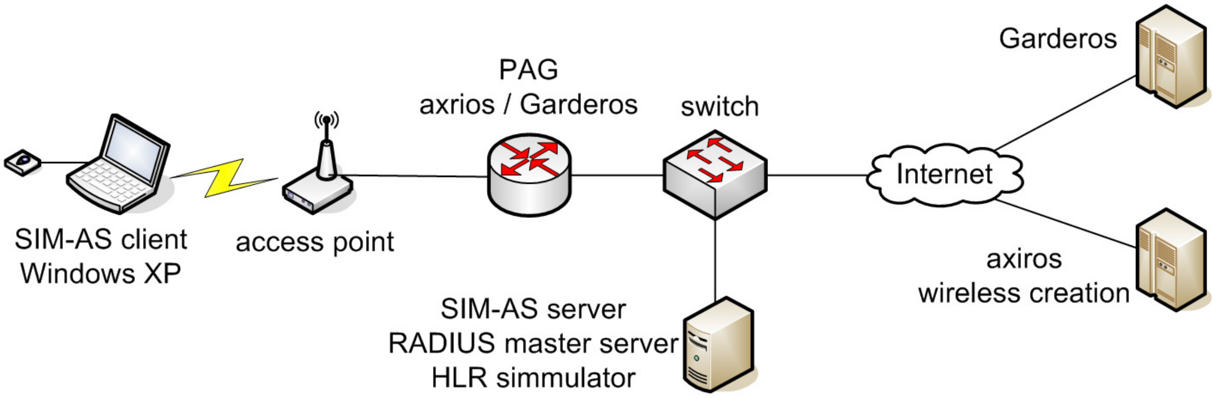


**Figure 5.13.** Test environment setup

# Chapter 6

# Evaluation

This chapter contains a discussion about the realized SIM-AS solution. It shows the solution from the users point of view, reviews the security provided by the solution and concludes with a summary.

## 6.1 Use case of SIM-AS

With the implemented protocol the user needs to find a wireless network that he wants to connect to. If more than one network is available he chooses one, an automatic selection of a SIM-AS network is not implemented. After he chooses a network he gets an IP address and other configuration information via DHCP. If he now starts a browser, he is presented the portal page of the local WISP operator. If the user starts his SIM-AS client and clicks on the login button, the SIM-AS client prompts the user for the PIN of his SIM card and then starts the authentication. During the authentication the user receives status updates. After the SIM-AS client displays successful authentication the user can just launch his browser and is able to surf the Internet.

## 6.2 Security considerations

As the solution is based on SIM cards even a Trojan horse cannot steal the necessary credentials for login. A Trojan horse may be able to intercept the PIN as it is entered into computer via the keyboard. But a misuse is still not possible without possession of the SIM card. It is impossible for a Trojan horse to extract all possible challenge responses as modern SIM cards have a limit of < 10000 lifetime triplet extraction.

For the purpose of this thesis the AEP-SIM protocol is considered sufficiently secure. EAP/SIM security is out of scope, interested readers are referated to [13] [31]. Therefore it can be assumed that the authentication itself is sufficiently secure. But what about session hijacking, certificates management, and weakness of local stored SSL certificates?

### 6.2.1 Session hijacking

The SIM-AS protocol only provides authentication and does not help in preventing IP or MAC address spoofing. As the SIM-AS protocol does not tunnel the client traffic it is not possible to prevent such attacks. Building a VPN to a machine within the WISP network is not possible due to the following problems:

- All traffic needs to be routed via the WISP network. The needed connection capacities are only available in a centralized WISP infrastructures.
- As shown in Subsection 4.7.1 VPN clients block all non-relevant layer3 traffic. Therefore it is not possible to use a VPN over a VPN.

Accordingly the session hijacking security has not been proofed if compared to web-based models.

### 6.2.2 Certificates management

In order for the SIM-AS client to verify the SIM-AS server certificate it needs a root certificate, which signed the SIM-AS server certificate. A problem arises if more than a few SIM-AS servers need to be deployed at a WISP. For each SIM-AS server a separate certificate is needed. This increases the work load for deploying SIM-AS servers. But using only one certificate for all SIM-AS servers causes the following problems:

- If one server is compromised all certificates need to be changed. Due to the update urgency a fast mechanism should be implemented before deploying the SIM-AS servers.
- The SIM-AS client needs to be aware of the multiple use of the certificate and needs to switch off the IP address / DNS name check for the certificate.

A provider also needs to sign every SIM-AS server of each roaming partner so that his SIM-AS clients are able to communicate with the foreign SIM-AS server. Furhtermore a way to revoke certificates is needed. A revoke system is not easily implemented and therefore an discussion, about the harm which can be done by an stolen private key, is necessary.

### 6.2.3 Weakness of local stored SSL certificates

The root certificate needs to be stored at the client. Therefore a Trojan horse can add an attacker certificate to the certificate list. Using this certificate it is possible to simulate a SIM-AS server by forwarding the EAP/SIM packets to the real server. The IP and MAC address which are send to the server can be exchanged with the addresses of the attacker. This attack is possible principle but an attacker needs to be at the same hotspot. Therefore he could as well just hijack a session.

## 6.3 Summary

The developed prototype shows that it is possible to build a solution, which fulfills following main requirements:

- SIM card based authentication
- No change at the access points
- Provide an interface to the EAP/SIM RADIUS server, which does not differ from the EAP/SIM solution
- Deployable in WISP backends
- Support roaming

It also highlights the problems which occur during the integration, for example the PAG ID. In summary the solution provides an easy way to upgrade existing WISPs to SIM based authentication.

The solution fulfills all requirements noted in section 4.1. But some problems remain open which implies that the solution may not be deployable under all circumstances:

- The problem with re-authentication and VPN client software is a principle one and cannot be solved with any layer3 protocol.
- The interface to the WISP system is the only point where integration work is necessary. This work cannot be done by Secartis. This is the main disatvantage of the solution. Every WISP system vendor needs to be convinced to integrate an proprietary interface to the SIM-AS solution. As seen during the development of the solution this is not that simple and it is time intensive. Chapter 7 shows a way out by removing this proprietary interface.

Some words about this thesis:

- Contacting the WISP system vendors early proofed to be good, as it took them some time to implement the prototype interface to the SIM-AS server.
- All used tools proofed their capableness and would be used again given the same circumstances.

# Chapter 7

# Outlook

The solution developed in this thesis can be easily adapted if integration in other scenarios is needed. Two of them are shown in this chapter.

## 7.1 SIM authentication without WISP support

If a company wants to provide their customers with the possibility to use the SIM card with roaming partners, which do not provide an integration interface for our solution, a different approach needs to be taken.

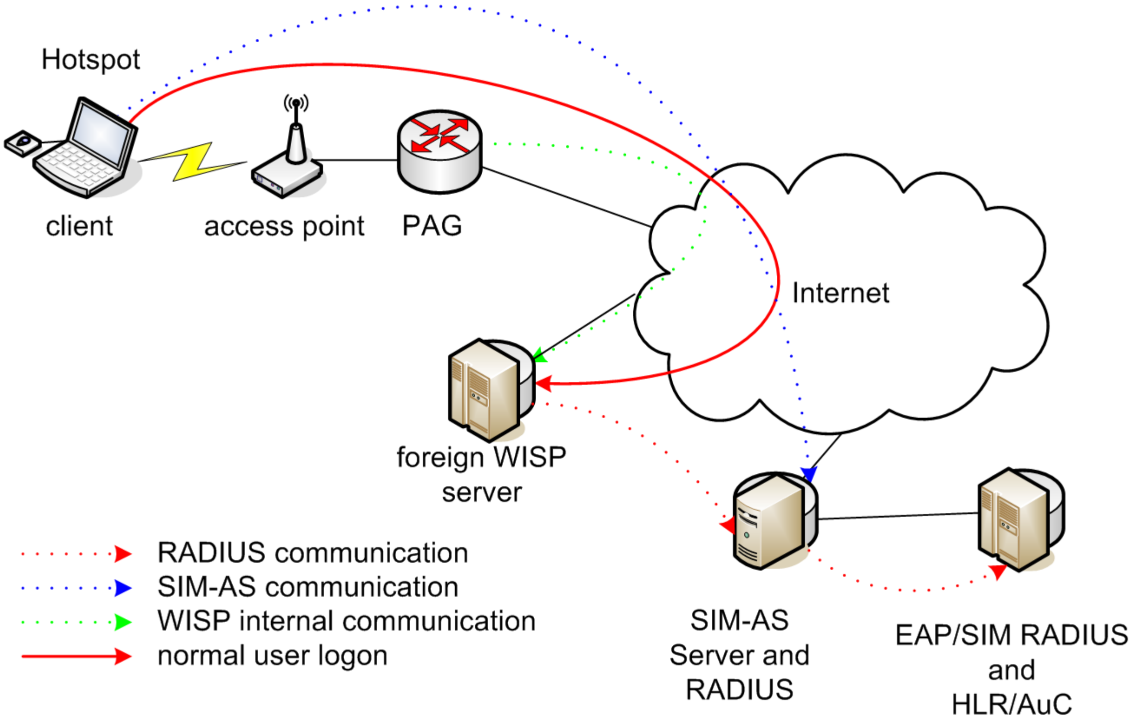


**Figure 7.1.** Typical setup for roaming without SIM-AS support from a roaming partner

The Wi-Fi Alliance has released a "Best Current Practices for Wireless Internet Service Provider (WISP) Roaming" document. This document describes the current best practice concerning roaming and also specifies an interface for the client to WISP interaction. This Universal Access Method (UAM) interface is based on the HTTP standard. Therefore it is possible for a subscriber to access WISP services with only a wireless network card and an Internet browser. But due the standardization it is also possible to use SmartClients to automatically login.

It is also based on RADIUS. Yet no RADIUS challenge is allowed. Therefore it is impossible to perform SIM authentication over it.

Figure 7.1 shows the setup of the environment and Figure 7.2 shows the sequence of commands for user login. Both show the same setup: a client connects to a foreign WISP and is still authenticated via SIM card. The authentication has following stages:

1. The SIM-AS server is within the walled garden of the foreign WISP and the SIM-AS client authenticates against the SIM-AS server. During this authentication the SIM-AS server communicates with the EAP/SIM enabled RADIUS server.
2. Within the RADIUS accept packet the RADIUS server provides the session encryption keys for the access point. The SIM-AS server takes these two values and generates a one time user name and password. With those the SIM-AS server creates a username and password database entry in a SQL database.

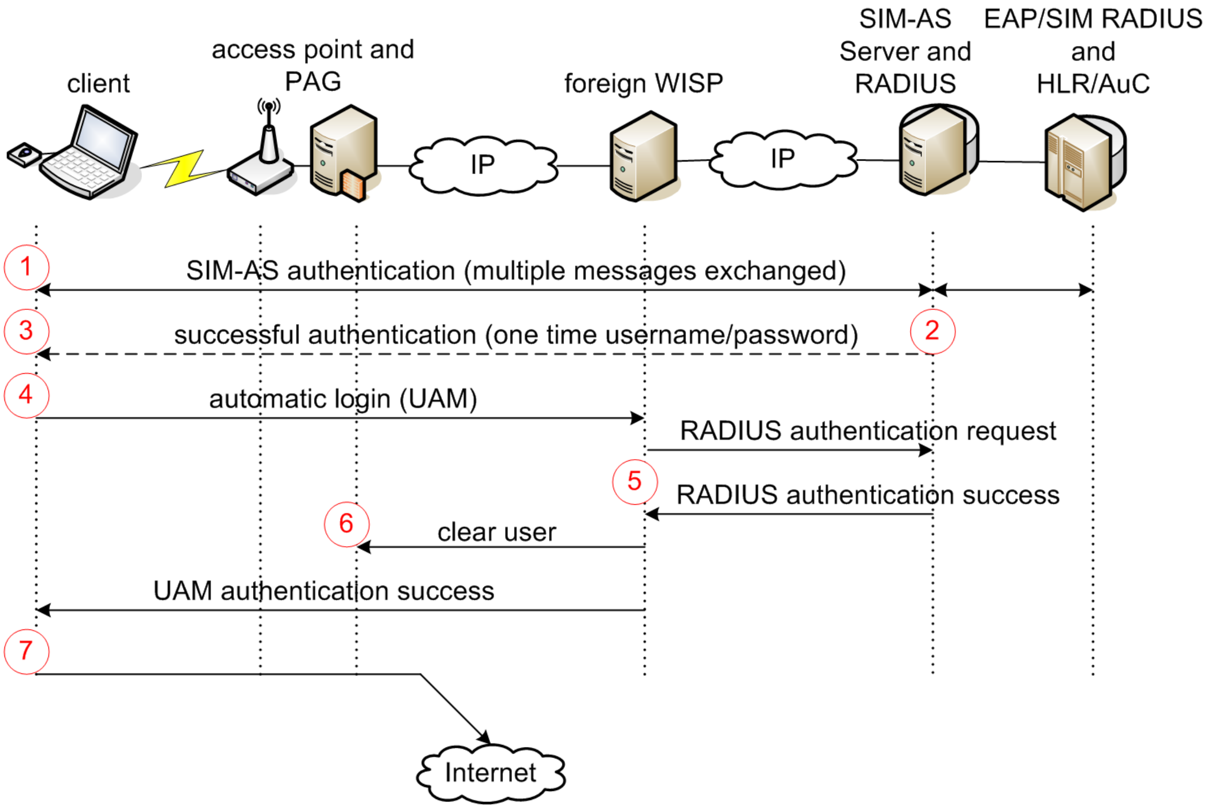

client
access point and PAG
foreign WISP
SIM-AS Server and RADIUS
EAP/SIM RADIUS and HLR/AuC
IP
IP
1
SIM-AS authentication (multiple messages exchanged)
3
successful authentication (one time username/password)
2
4
automatic login (UAM)
RADIUS authentication request
5
RADIUS authentication success
6
clear user
UAM authentication success
7
Internet

**Figure 7.2.** Authentication chain for roaming without SIM-AS support from a roaming partner

3. The client calculates its session keys and derives a one time user name and password combination.
4. Via UAM the client sends the onetime tokens to the foreign WISP server, which requests, as configured for the client used realm, an authentication from the SIM-AS RADIUS server.
5. This RADIUS server looks up the combination in its SQL database, and sends an accept or reject.
6. The WISP server informs the PAG via a WISP system proprietary protocol to clear the client.
7. The client receives the success or failure of the authentication and informs the user accordingly.

With the above delineated solution it is possible to support SIM authentication for customers even in the case of non-SIM-AS roaming partners.

## 7.2 SIM authentication for web content

As the SIM-AS client is written with Java it is possible to use the authentication for web content billing. This is especially interesting if the customer already has a SIM card and reader. Figure 7.3 shows a possible setup:

1. The client visits a public Internet URL and clicks on a so called Premium URL. The user

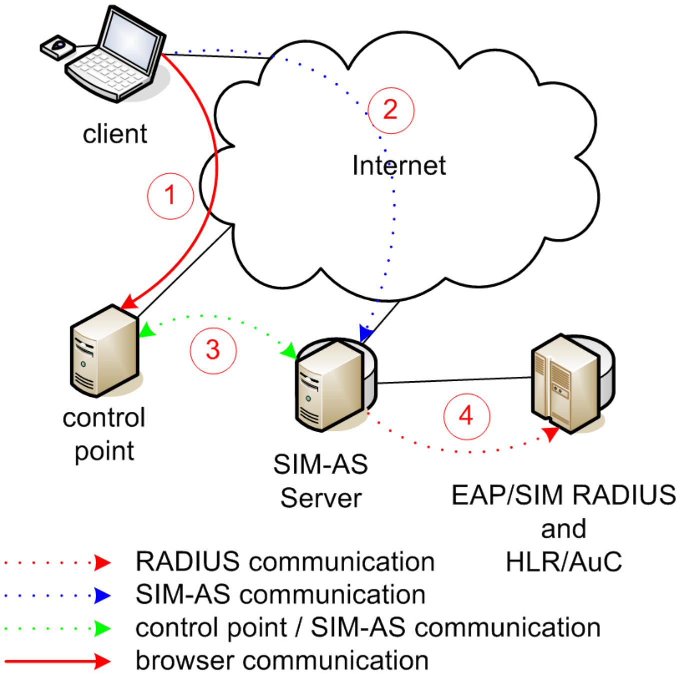


**Figure 7.3.** Typical setup for authenticate function for web content

is redirected to ControlPoint Server, which shows an HTML page with our SIM-AS client as plugin. The applet is provided with a session ID by the ControlPoint server.

2. After the plugin is download and launched on the client, it starts an authentication to the SIM-AS server. The client sends the session ID issued by the ControlPoint server with the authentication packets to the SIM-AS server.

3. The SIM-AS server informs the ControlPoint server that the user with session ID is authenticated with him.

4. This is our normal RADIUS connection from the SIM-AS to the EAP/SIM enabled RADIUS server.

This solution provides an anonymous and secure authentication method for Premium content.

## 7.3 Open questions

Following issues were not terminally solved and need to be discussed further.

1. The protocol EAP/SIM is still no RFC yet. There exists the slight chance that protocol will be rejected. It is also possible that some adjustments need to be done.

2. During this thesis some assumptions were conducted. These worked well for our prototype, but it still needs to be shown if they also work in a production environment. Especially the PAG identification solutions need to prove their fitnesses in a real live environment.

3. No usability tests were conducted with the implemented prototype. The prototype was only operated by IT professionals so far. Before deploying the solution, the user interface should be verified against user interface guidelines.

4. The SIM-AS client is written in Java. For a non prototype version an install method should be developed, which does not require the JDK 1.4.2 to be installed.

# Glossary

### A

**active logout:** An active logout is initiated by the user and requires the software client to transmit the wish to a server of the WISP. compare passive logout

**authenticator:** The end of the EAP link initiating EAP authentication

### H

**hotspot:** A hotspot is a WLAN access point or area, in particular for connecting to Internet

### I

**IEEE:** The IEEE (Eye-triple-E) is a non-profit, technical professional association of more than 360,000 individual members in approximately 175 countries. The full name is the Institute of Electrical and Electronics Engineers, Inc., although the organization is most popularly known and referred to by the letters I-E-E-E.

### P

**passive logout:** A logout which occurs under certain circumstances like idle timeout or maximum session time. compare active logout

**peer:** The end of the EAP link that responds to the authenticator

### R

**roaming:** Roaming is a general term in wireless telecommunications that refers to the extending of connectivity service in a network that is different than the network with which a station is registered.The canonical example of "roaming" is for cellular phones, when you take your phone to an area where your service provider does not have coverage [10].

### S

**SIM:** A subscriber identity module (SIM) is a smart card securely storing the key identifying a mobile subscriber. SIMs are most widely used in GSM systems, but a compatible module is also used for UMTS UEs. The card also contains storage space for text messages and a phone book [10].

**supplicant:** The end of the EAP link that responds to the authenticator

### W

**walled garden:** A walled garden, with regards to content, refers to an exclusive set of information services provided for users.

This is in contrast to consumers going to the open Internet for content and e-commerce. The term is often used to describe offerings from interactive television providers or mobile phone operators which provide custom content, and not simply common carrier functions [10]. Within this master thesis walled garden is only used to describe servers and services which can be reached by the client prior authentication.

**WISP:** WISP is an acronym which stands for Wireless Internet Service Provider

**WLAN:** Wireless Local Area Network

# Bibliography

[1] ANSI/IEEE. *ISO/IEC 8802-11:1999/Amd 1:2000(E), Part 11: Wireless LAN Medium Access Control(MAC) and Physical Layer (PHY) specifications: Amendment 1: High-speed Physical Layer in the 5GHz band*. ISO/IEC, January 1999.

[2] ANSI/IEEE. *ISO/IEC 8802-11:1999(E), Part 11: Wireless LAN Medium Access Control(MAC) and Physical Layer (PHY) specifications*. ISO/IEC, January 1999.

[3] SCT Yvon Avenel. Sim card is a prerequisite for wi-fi access. http://www.smartcardstrends.com/det_atc.php?idu=5, October 2003. [visited at 14. July 2004] .

[4] B. Aboba. *RFC3575: IANA Considerations for RADIUS (Remote Authentication Dial In User Service)*. Internet Engineering Task Force, ftp://ftp.rfc-editor.org/in-notes/rfc3575.txt, July 2003.

[5] B. Aboba, J. Vollbrecht. *RFC2607: Proxy Chaining and Policy Implementation in Roaming*. Internet Engineering Task Force, ftp://ftp.rfc-editor.org/in-notes/rfc2607.txt, June 1999.

[6] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, H. Levkowetz, Ed. *RFC3748: Extensible Authentication Protocol (EAP)*. Internet Engineering Task Force, ftp://ftp.rfc-editor.org/in-notes/rfc3748.txt, June 2004.

[7] B. Aboba, P. Calhoun. *RFC3579: RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP)*. Internet Engineering Task Force, ftp://ftp.rfc-editor.org/in-notes/rfc3579.txt, September 2003.

[8] C. Rigney, S. Willens, A. Rubens, W. Simpson. *RFC2865: Remote Authentication Dial In User Service (RADIUS)*. Internet Engineering Task Force, ftp://ftp.rfc-editor.org/in-notes/rfc2865.txt, June 2000.

[9] Bruno Chiarelli. Secartis: Erstes kommerziell eingesetztes wlan-authentisierungssystem über eap sim. http://www.itseccity.de/?url=/content/markt/invests/031022_mar_inv_secartis.html, October 2003. [visited at 14. July 2004] .

[10] Wikipedia Community. Wikipedia: The free encylopedia. http://www.wikipedia.org/, July 2004. [visited at 14. July 2004] .

[11] D. Mitton. *RFC2882: Network Access Servers Requirements: Extended RADIUS Practices*. Internet Engineering Task Force, ftp://ftp.rfc-editor.org/in-notes/rfc2882.txt, July 2000.

[12] ETSI. *ETSI TS 102 310 v2.0.0 Smart Cards; Extensible Authentication Protocol support in the UICC; (Release 6)*. European Telecommunication Standard Institute, May 2004.

[13] H. Haverinen, J. Salowey. *Extensible Authentication Protocol Method for GSM Subscriber Identity Modules (EAP-SIM) Draft 13*. Internet Engineering Task Force, http://www.ietf.org/internet-drafts/draft-haverinen-pppext-eap-sim-13.txt, April 2004.

[14] Uwe Hübner. Wireless local area networks. `http://www.york.ac.uk/depts/maths/histstat/lifework.htm`, July 2004. [visited at 13. July 2004] .

[15] Firoz Kaderali. Sicherheit in gsm-netzen. `http://www.et-online.fernuni-hagen.de/lehre/etk007/`, July 2004. [visited at 13. July 2004] .

[16] Jürgen Kuri. Drahtlose netze als alternative zum kabelgebundenen ethernet. *c't - Magazin für Computertechnik*, 25:132, 1999.

[17] LAN MAN Standards Committee of the IEEE Computer Society. *IEEE Std 802.11-1997, Part 11: Wireless LAN Medium Access Control(MAC) and Physical Layer (PHY) specifications*. IEEE, New York, June 1997.

[18] LAN MAN Standards Committee of the IEEE Computer Society. *IEEE Std 802.11a-1999, Part 11: Wireless LAN Medium Access Control(MAC) and Physical Layer (PHY) specifications: Amendment 1: High-speed Physical Layer in the 5GHz ban*. IEEE, New York, September 1999.

[19] LAN MAN Standards Committee of the IEEE Computer Society. *IEEE Std 802.11b-1999, Part 11: Wireless LAN Medium Access Control(MAC) and Physical Layer (PHY) specifications: Amendment 2: Higher-speed Physical Extension Layer in the 2,4GHz Band*. IEEE, New York, September 1999.

[20] LAN MAN Standards Committee of the IEEE Computer Society. *IEEE Std 802.11b-1999/Cor 1-2001, Part 11: Wireless LAN Medium Access Control(MAC) and Physical Layer (PHY) specifications: Amendment 2: Higher-speed Physical Extension Layer in the 2,4GHz Band Corrigendum 1*. IEEE, New York, November 2001.

[21] LAN MAN Standards Committee of the IEEE Computer Society. *IEEE Std 802.11d-2001, Part 11: Wireless LAN Medium Access Control(MAC) and Physical Layer (PHY) specifications: Amendment 3: Specification for operation in additional regulatory domains*. IEEE, New York, November 2001.

[22] LAN MAN Standards Committee of the IEEE Computer Society. *IEEE Std 802.1x-2001, Local and metropolitan area networks - Port-Based Network Access Control*. IEEE, New York, June 2001.

[23] LAN MAN Standards Committee of the IEEE Computer Society. *IEEE Std 802.11f-2003, IEEE Trial-Use Recommended Practice for Multi-Vendor Access Point Interoperability*

*via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation*. IEEE, New York, July 2003.

[24] LAN MAN Standards Committee of the IEEE Computer Society. *IEEE Std 802.11g-2003, Part 11: Wireless LAN Medium Access Control(MAC) and Physical Layer (PHY) specifications: Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band*. IEEE, New York, June 2003.

[25] LAN MAN Standards Committee of the IEEE Computer Society. *IEEE Std 802.11h-2003, Part 11: Wireless LAN Medium Access Control(MAC) and Physical Layer (PHY) specifications: Amendment 5: Spectrum and Transmit Power Management Extensions in the 5 Ghz band in Europe*. IEEE, New York, October 2003.

[26] Richard Linke. Hotspot-betreiber in der zange. *c't - Magazin für Computertechnik*, 14:90, 2004.

[27] M. Chiba, G. Dommety, M. Eklund, D. Mitton, B. Aboba. *RFC3576: Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS)*. Internet Engineering Task Force, ftp://ftp.rfc-editor.org/in-notes/rfc3576.txt, July 2003.

[28] Günter May. Radius-server für novell netware. http://www.lrz-muenchen.de/services/netzdienste/modem-isdn/radiusnds/, February 2001. [visited at 13. July 2004] .

[29] mp technology consulting GmbH. Hotspot locations - the wireless directory. http://www.hotspot-locations.com, July 2004. [visited at 07. July 2004] .

[30] P. Congdon, B. Aboba, A. Smith, G. Zorn, J. Roese. *RFC3580: IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines*. Internet Engineering Task Force, ftp://ftp.rfc-editor.org/in-notes/rfc3580.txt, September 2003.

[31] Sarva Patel. Analysis of eap-sim session key agreement. http://www.drizzle.com/~aboba/EAP/AnalyisOfEAP.pdf. [visited at 14. July 2004] .

[32] Malte Schmidt-Tychsen Patrick Brauch. Wlan-verschlüsselung wep ist unsicher. *c't - Magazin für Computertechnik*, 19:39, 2001.

[33] R. Housley, T. Moore. *RFC3770: Certificate Extensions and Attributes Supporting Authentication in Point-to-Point Protocol (PPP) and Wireless Local Area Networks (WLAN)*. Internet Engineering Task Force, ftp://ftp.rfc-editor.org/in-notes/rfc3770.txt, May 2004.

[34] James L. Massey Rainer A. Rueppel. *The Security of Natel D GSM*. Berner Technologie-Forum on Mobile Communications, 1991.

[35] Inga Rapp. Greenspot soll hotspot-surfen vereinheitlichen. *c't - Magazin für Computertechnik*, 09:44, 2003.

[36] Matthias K. Weber Siegmund H. Redl. *An introduction to GSM*. Artech House, Malcolm W. Oliphant Boston, 1995.

[37] Universität Rostock, FB Informatik, Lehrstuhl Rechnerarchitektur. Wlan standards. `http://wlan.informatik.uni-rostock.de/standard/`. [visited at 14. July 2004] .

[38] Heise Zeitschriften Verlag. c't-browsercheck. `http://www.heise.de/security/dienste/browsercheck/`, July 2004. [visited at 12. July 2004] .